

MTP_BNN

December 22, 2021

packages

[1]: `#empty`

[2]: `!pip install pyforest`

```
Collecting pyforest
  Downloading pyforest-1.1.0.tar.gz (15 kB)
Building wheels for collected packages: pyforest
  Building wheel for pyforest (setup.py) ... done
  Created wheel for pyforest: filename=pyforest-1.1.0-py2.py3-none-any.whl
size=14607
sha256=4d75a2c0e796b4f628cf8ba1072bab644f149970343662d6913e7e4e693219f8
  Stored in directory: /root/.cache/pip/wheels/61/1c/da/48e6c884142d485475d852d6
9d20a096aba5beceb338822893
Successfully built pyforest
Installing collected packages: pyforest
Successfully installed pyforest-1.1.0
```

[3]: `#automatic imports required packages as per usage in code`
`import pyforest`

[4]: `#packages`
`!pip install tensorflow-probability`
`!pip install nbconvert`

```
Requirement already satisfied: tensorflow-probability in
/usr/local/lib/python3.7/dist-packages (0.15.0)
Requirement already satisfied: dm-tree in /usr/local/lib/python3.7/dist-packages
(from tensorflow-probability) (0.1.6)
Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-
packages (from tensorflow-probability) (4.4.2)
Requirement already satisfied: cloudpickle>=1.3 in /usr/local/lib/python3.7
/dist-packages (from tensorflow-probability) (1.3.0)
Requirement already satisfied: numpy>=1.13.3 in /usr/local/lib/python3.7/dist-
packages (from tensorflow-probability) (1.19.5)
Requirement already satisfied: gast>=0.3.2 in /usr/local/lib/python3.7/dist-
packages (from tensorflow-probability) (0.4.0)
```

Requirement already satisfied: absl-py in /usr/local/lib/python3.7/dist-packages (from tensorflow-probability) (0.12.0)

Requirement already satisfied: six>=1.10.0 in /usr/local/lib/python3.7/dist-packages (from tensorflow-probability) (1.15.0)

Requirement already satisfied: nbconvert in /usr/local/lib/python3.7/dist-packages (5.6.1)

Requirement already satisfied: nbformat>=4.4 in /usr/local/lib/python3.7/dist-packages (from nbconvert) (5.1.3)

Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from nbconvert) (5.1.1)

Requirement already satisfied: entrypoints>=0.2.2 in /usr/local/lib/python3.7/dist-packages (from nbconvert) (0.3)

Requirement already satisfied: testpath in /usr/local/lib/python3.7/dist-packages (from nbconvert) (0.5.0)

Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from nbconvert) (2.6.1)

Requirement already satisfied: jupyter-core in /usr/local/lib/python3.7/dist-packages (from nbconvert) (4.9.1)

Requirement already satisfied: defusedxml in /usr/local/lib/python3.7/dist-packages (from nbconvert) (0.7.1)

Requirement already satisfied: jinja2>=2.4 in /usr/local/lib/python3.7/dist-packages (from nbconvert) (2.11.3)

Requirement already satisfied: bleach in /usr/local/lib/python3.7/dist-packages (from nbconvert) (4.1.0)

Requirement already satisfied: mistune<2,>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert) (0.8.4)

Requirement already satisfied: pandocfilters>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from nbconvert) (1.5.0)

Requirement already satisfied: MarkupSafe>=0.23 in /usr/local/lib/python3.7/dist-packages (from jinja2>=2.4->nbconvert) (2.0.1)

Requirement already satisfied: jsonschema!=2.5.0,>=2.4 in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.4->nbconvert) (2.6.0)

Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.7/dist-packages (from nbformat>=4.4->nbconvert) (0.2.0)

Requirement already satisfied: packaging in /usr/local/lib/python3.7/dist-packages (from bleach->nbconvert) (21.3)

Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-packages (from bleach->nbconvert) (1.15.0)

Requirement already satisfied: webencodings in /usr/local/lib/python3.7/dist-packages (from bleach->nbconvert) (0.5.1)

Requirement already satisfied: pyparsing!=3.0.5,>=2.0.2 in /usr/local/lib/python3.7/dist-packages (from packaging->bleach->nbconvert) (3.0.6)

```
[5]: import pandas as pd
import numpy as np
```

0.0.1 DATA

import data

```
[6]: #using official url to load data
url = 'https://archive.ics.uci.edu/ml/machine-learning-databases/00601/ai4i2020.
      ↪CSV'

data = pd.read_csv(url)

data.head()
```

```
[6]:   UDI Product ID Type  Air temperature [K]  ...  HDF  PWF  OSF  RNF
0     1      M14860    M           298.1  ...    0    0    0    0
1     2      L47181    L           298.2  ...    0    0    0    0
2     3      L47182    L           298.1  ...    0    0    0    0
3     4      L47183    L           298.2  ...    0    0    0    0
4     5      L47184    L           298.2  ...    0    0    0    0
```

[5 rows x 14 columns]

data description taken from UCI:

Abstract: The AI4I 2020 Predictive Maintenance Dataset is a synthetic dataset that reflects real predictive maintenance data encountered in industry.

Variable	Value
Data Set Characteristics:	Multivariate, Time-Series
Number of Instances:	10000
Area:	Computer
Attribute Characteristics:	Real
Number of Attributes:	14
Date Donated:	2020-08-30
Associated Tasks:	Classification, Regression, Causal-Discovery
Missing Values?	N/A
Number of Web Hits:	33135

** Data Set Information: **

Since real predictive maintenance datasets are generally difficult to obtain and in particular difficult to publish, we present and provide a synthetic dataset that reflects real predictive maintenance encountered in industry to the best of our knowledge.

Attribute Information:

The dataset consists of 10 000 data points stored as rows with 14 features in columns UID: unique identifier ranging from 1 to 10000 product ID: consisting of a letter L, M, or H for low (50% of all products), medium (30%) and high (20%) as product quality variants and a variant-specific serial number air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K. rotational speed [rpm]: calculated from a power of 2860 W, overlaid with a normally distributed

noise torque [Nm]: torque values are normally distributed around 40 Nm with a $\sigma = 10$ Nm and no negative values. tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process. and a 'machine failure' label that indicates, whether the machine has failed in this particular datapoint for any of the following failure modes are true.

The machine failure consists of five independent failure modes tool wear failure (TWF): the tool will be replaced or fail at a randomly selected tool wear time between 200 – 240 mins (120 times in our dataset). At this point in time, the tool is replaced 69 times, and fails 51 times (randomly assigned). heat dissipation failure (HDF): heat dissipation causes a process failure, if the difference between air- and process temperature is below 8.6 K and the tool's rotational speed is below 1380 rpm. This is the case for 115 data points. power failure (PWF): the product of torque and rotational speed (in rad/s) equals the power required for the process. If this power is below 3500 W or above 9000 W, the process fails, which is the case 95 times in our dataset. overstrain failure (OSF): if the product of tool wear and torque exceeds 11,000 minNm for the L product variant (12,000 M, 13,000 H), the process fails due to overstrain. This is true for 98 datapoints. random failures (RNF): each process has a chance of 0,1 % to fail regardless of its process parameters. This is the case for only 5 datapoints, less than could be expected for 10,000 datapoints in our dataset.

If at least one of the above failure modes is true, the process fails and the 'machine failure' label is set to 1. It is therefore not transparent to the machine learning method, which of the failure modes has caused the process to fail

Relevant Papers:

Stephan Matzka, 'Explainable Artificial Intelligence for Predictive Maintenance Applications', Third International Conference on Artificial Intelligence for Industries (AI4I 2020), 2020 (in press)

```
[7]: data.describe()
```

```
[7]:
```

	UDI	Air temperature [K]	...	OSF	RNF
count	10000.00000	10000.000000	...	10000.000000	10000.00000
mean	5000.50000	300.004930	...	0.009800	0.00190
std	2886.89568	2.000259	...	0.098514	0.04355
min	1.00000	295.300000	...	0.000000	0.00000
25%	2500.75000	298.300000	...	0.000000	0.00000
50%	5000.50000	300.100000	...	0.000000	0.00000
75%	7500.25000	301.500000	...	0.000000	0.00000
max	10000.00000	304.500000	...	1.000000	1.00000

[8 rows x 12 columns]

```
[8]: #for i in data:
      #print(data[i].unique())
```

```
[9]: data.nunique()
```

```
[9]:
```

UDI	10000
Product ID	10000
Type	3
Air temperature [K]	93
Process temperature [K]	82
Rotational speed [rpm]	941
Torque [Nm]	577

```

Tool wear [min]          246
Machine failure          2
TWF                      2
HDF                      2
PWF                      2
OSF                      2
RNF                      2
dtype: int64

```

[10]: *#basic info about dataset*

```

df = data
df.shape
df.index
df.columns
df.info()
df.count()

```

```
<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 10000 entries, 0 to 9999
```

```
Data columns (total 14 columns):
```

#	Column	Non-Null Count	Dtype
0	UDI	10000 non-null	int64
1	Product ID	10000 non-null	object
2	Type	10000 non-null	object
3	Air temperature [K]	10000 non-null	float64
4	Process temperature [K]	10000 non-null	float64
5	Rotational speed [rpm]	10000 non-null	int64
6	Torque [Nm]	10000 non-null	float64
7	Tool wear [min]	10000 non-null	int64
8	Machine failure	10000 non-null	int64
9	TWF	10000 non-null	int64
10	HDF	10000 non-null	int64
11	PWF	10000 non-null	int64
12	OSF	10000 non-null	int64
13	RNF	10000 non-null	int64

```
dtypes: float64(3), int64(9), object(2)
```

```
memory usage: 1.1+ MB
```

```

[10]: UDI          10000
      Product ID   10000
      Type         10000
      Air temperature [K] 10000
      Process temperature [K] 10000
      Rotational speed [rpm] 10000
      Torque [Nm]      10000
      Tool wear [min]  10000
      Machine failure  10000

```

```
TWF          10000
HDF          10000
PWF          10000
OSF          10000
RNF          10000
dtype: int64
```

```
[11]: df.sum()
df.cumsum()
df.min()
df.max()
df.describe()
df.mean()
df.median()
```

```
[11]: UDI          5000.5
Air temperature [K]    300.1
Process temperature [K] 310.1
Rotational speed [rpm] 1503.0
Torque [Nm]           40.1
Tool wear [min]       108.0
Machine failure        0.0
TWF                   0.0
HDF                   0.0
PWF                   0.0
OSF                   0.0
RNF                   0.0
dtype: float64
```

preprocessing data

```
[12]: #define X and y from df
# product id is unique for each data row and its not important
# but we have product type of 3 categories
# L, M, H are three types representing for low (50% of all products),
# medium (30%) and high (20%) as product quality variants respectively
df['Type'].unique()
```

```
[12]: array(['M', 'L', 'H'], dtype=object)
```

```
[13]: # converting this categorical data to numerical with class 0, 1, 2 for L,M,H
      →respectively
# using OrdinalEncoder from sklearn for ordinal data of product quality variant
# indicating l for low quality, m for medium quality, h for high quality
# one-hot encoding is not suitable for ordinal data
from sklearn.preprocessing import OrdinalEncoder
ordinal_encoder = OrdinalEncoder()
df['Type'] = ordinal_encoder.fit_transform(df[['Type']])
df['Type'].unique()
```

```
# this gives categories converted into integers
```

```
[13]: array([2., 1., 0.])
```

```
[14]: # these are original categories in data
ordinal_encoder.categories_
```

```
[14]: [array(['H', 'L', 'M'], dtype=object)]
```

```
[15]: # this sorts all the categories present and assigns values to them in
      ↪ alphabetical order
```

```
# 0 for H
# 1 for L
# 2 for M
print(ordinal_encoder.inverse_transform([[0]]))
print(ordinal_encoder.inverse_transform([[1]]))
print(ordinal_encoder.inverse_transform([[2]]))
```

```
[['H']]
```

```
[['L']]
```

```
[['M']]
```

```
[16]: df.describe()
```

```
[16]:
```

	UDI	Type	...	OSF	RNF
count	10000.00000	10000.00000	...	10000.000000	10000.00000
mean	5000.50000	1.19940	...	0.009800	0.00190
std	2886.89568	0.60023	...	0.098514	0.04355
min	1.00000	0.00000	...	0.000000	0.00000
25%	2500.75000	1.00000	...	0.000000	0.00000
50%	5000.50000	1.00000	...	0.000000	0.00000
75%	7500.25000	2.00000	...	0.000000	0.00000
max	10000.00000	2.00000	...	1.000000	1.00000

```
[8 rows x 13 columns]
```

```
[17]: df.nunique()
```

```
[17]:
```

UDI	10000
Product ID	10000
Type	3
Air temperature [K]	93
Process temperature [K]	82
Rotational speed [rpm]	941
Torque [Nm]	577
Tool wear [min]	246
Machine failure	2
TWF	2
HDF	2
PWF	2
OSF	2

RNF

2

dtype: int64

[18]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   UDI                                    10000 non-null  int64
1   Product ID                           10000 non-null  object
2   Type                                  10000 non-null  float64
3   Air temperature [K]                  10000 non-null  float64
4   Process temperature [K]              10000 non-null  float64
5   Rotational speed [rpm]               10000 non-null  int64
6   Torque [Nm]                          10000 non-null  float64
7   Tool wear [min]                      10000 non-null  int64
8   Machine failure                      10000 non-null  int64
9   TWF                                  10000 non-null  int64
10  HDF                                  10000 non-null  int64
11  PWF                                  10000 non-null  int64
12  OSF                                  10000 non-null  int64
13  RNF                                  10000 non-null  int64
dtypes: float64(4), int64(9), object(1)
memory usage: 1.1+ MB
```

```
[19]: # now make the final dataset to be used in NN
# remove the product id variable
# remaining attributes are of types either int64 or float64
df.drop('Product ID', axis=1, inplace=True)
df.drop('UDI', axis=1, inplace=True)
```

[20]: df.info()

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 12 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   Type                                  10000 non-null  float64
1   Air temperature [K]                  10000 non-null  float64
2   Process temperature [K]              10000 non-null  float64
3   Rotational speed [rpm]               10000 non-null  int64
4   Torque [Nm]                          10000 non-null  float64
5   Tool wear [min]                      10000 non-null  int64
6   Machine failure                      10000 non-null  int64
7   TWF                                  10000 non-null  int64
```



```

8   HDF          10000 non-null int64
9   PWF          10000 non-null int64
10  OSF          10000 non-null int64
11  RNF          10000 non-null int64
dtypes: float64(4), int64(8)
memory usage: 937.6 KB

```

```
[21]: ## add mitosheet data visualization
```

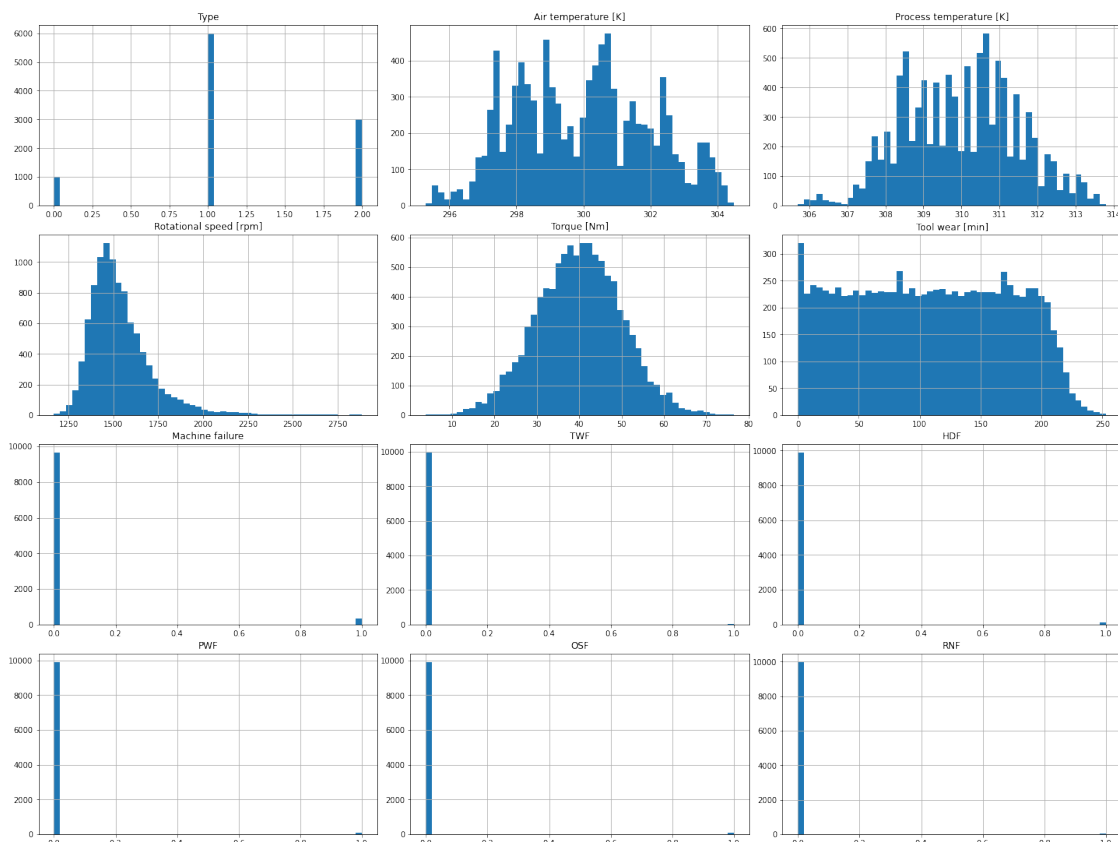
mitosheet visualization code

```
[22]: # exploring data
```

```
[23]: df.hist(bins=50, figsize=(20,15))
plt.tight_layout(pad=0.4)
plt.show()
```

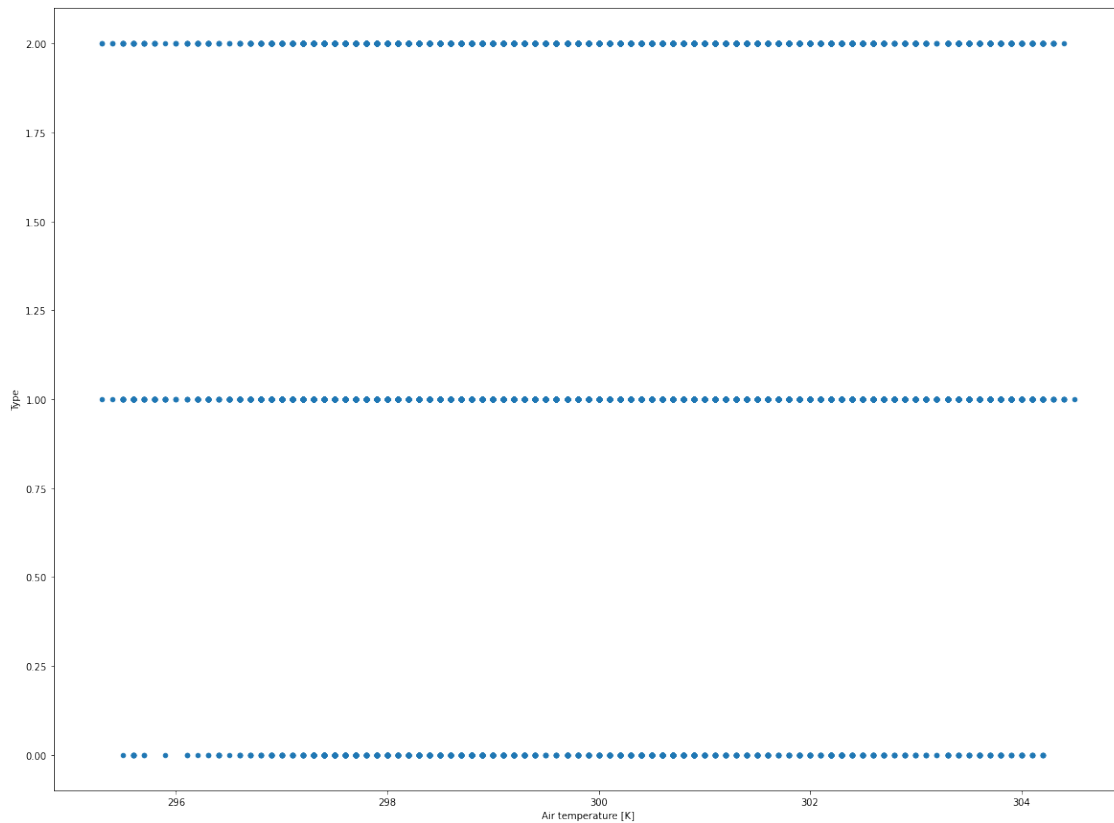
<IPython.core.display.Javascript object>

<IPython.core.display.Javascript object>



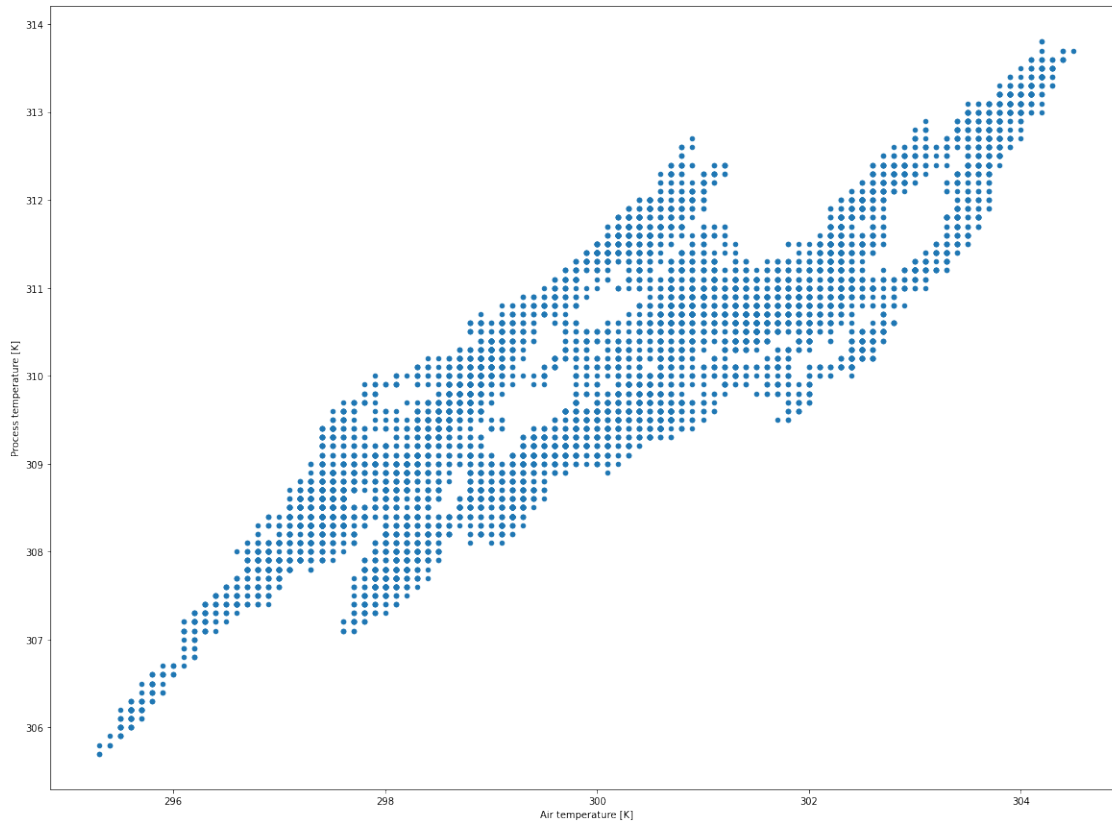
```
[24]: df.plot.scatter(y = 'Type',x='Air temperature [K]', figsize=(20,15))  
plt.show()
```

<IPython.core.display.Javascript object>



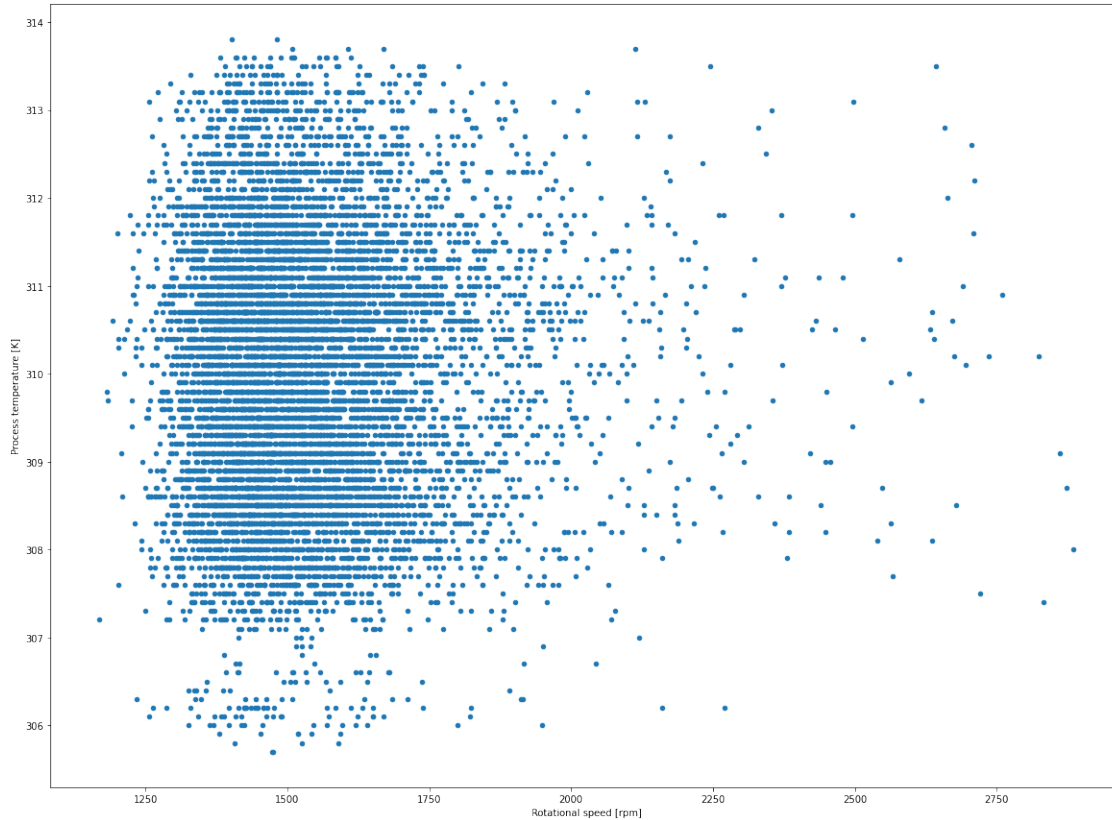
```
[25]: df.plot.scatter(y = 'Process temperature [K]',x='Air temperature [K]',  
    ↳figsize=(20,15))  
plt.show()
```

<IPython.core.display.Javascript object>



```
[26]: df.plot.scatter(y = 'Process temperature [K]',x='Rotational speed [rpm]',  
    ↪figsize=(20,15))  
plt.show()
```

<IPython.core.display.Javascript object>



[27]: `#confusion matrix`

[28]: `import seaborn as sns`

```
#correlation matrix
numeric_col = ['Type', 'Air temperature [K]', 'Process temperature [K]',
               'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]',
               'TWF', 'HDF', 'PWF', 'OSF', 'RNF', 'Machine failure']
# Correlation Matrix formation
corr_matrix = df.loc[:,numeric_col].corr()
print(corr_matrix)
#Using heatmap to visualize the correlation matrix
fig, ax = plt.subplots(figsize=(10,10))
sns.heatmap(corr_matrix, annot=True,ax=ax)
```

	Type	...	Machine failure
Type	1.000000	...	-0.005152
Air temperature [K]	0.017599	...	0.082556
Process temperature [K]	0.013444	...	0.035946
Rotational speed [rpm]	-0.002693	...	-0.044188
Torque [Nm]	0.004011	...	0.191321
Tool wear [min]	-0.003930	...	0.105448

```

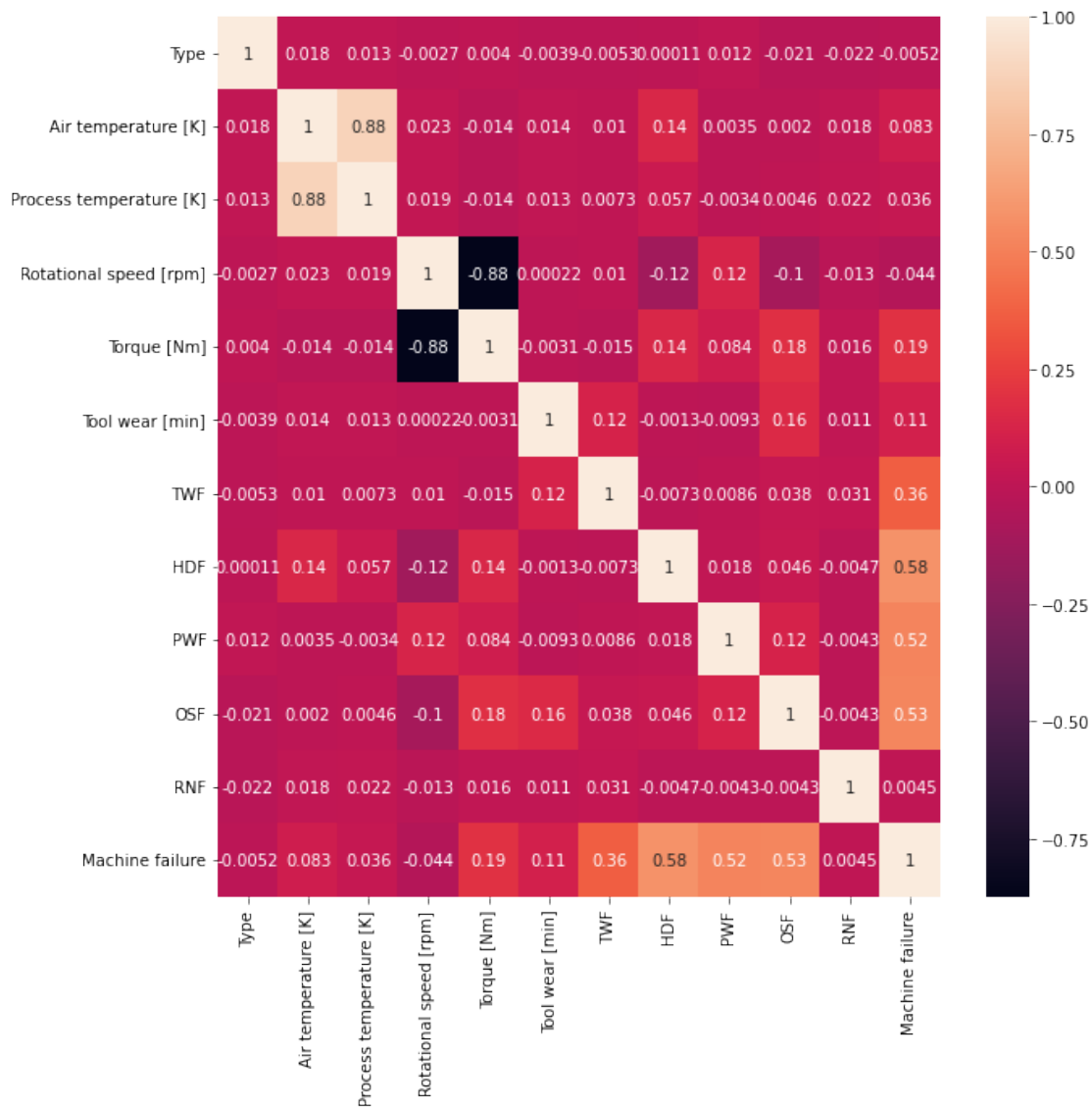
TWF          -0.005349 ...      0.362904
HDF           0.000108 ...      0.575800
PWF           0.012121 ...      0.522812
OSF          -0.021211 ...      0.531083
RNF          -0.022147 ...      0.004516
Machine failure -0.005152 ...      1.000000

```

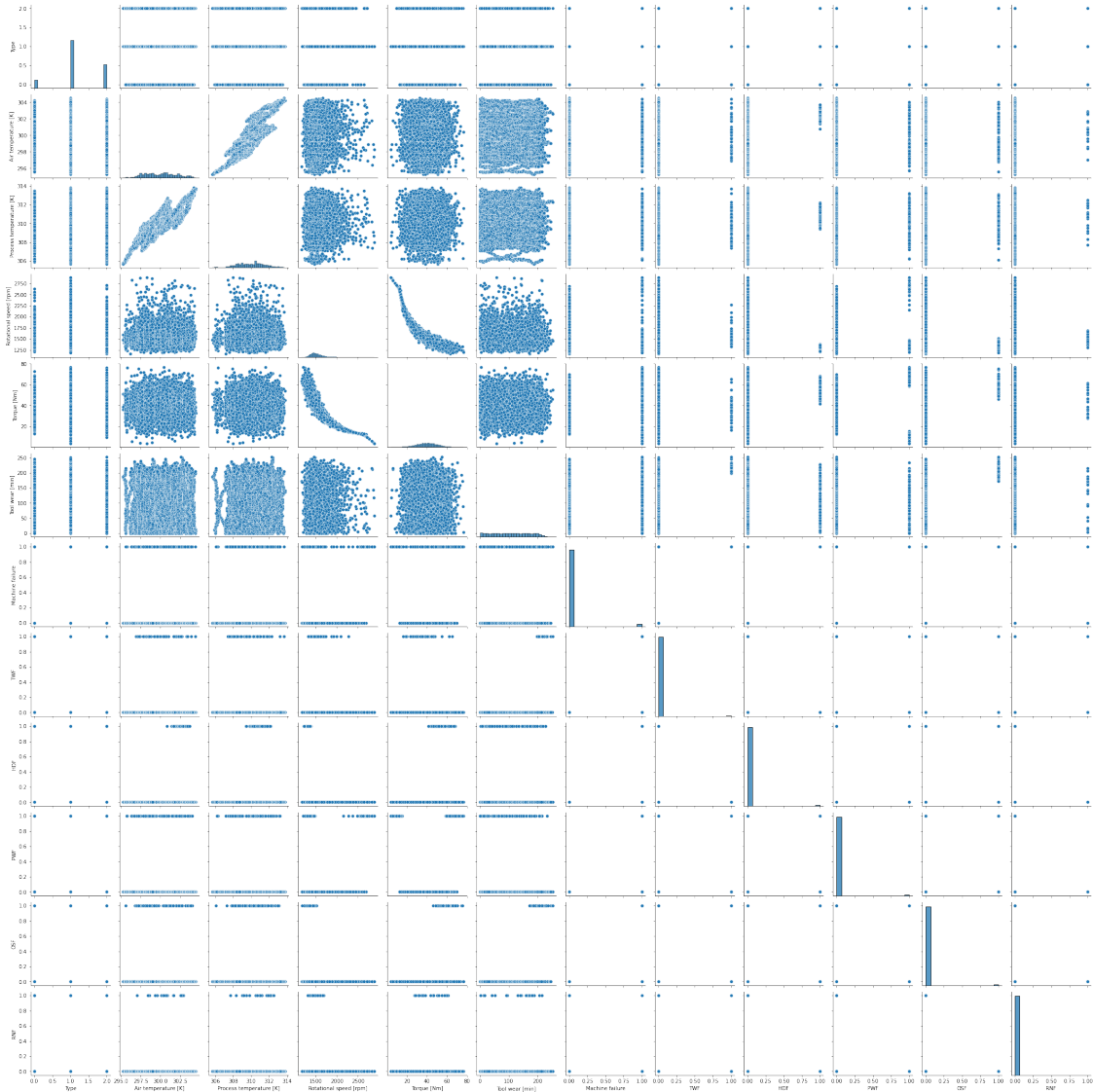
[12 rows x 12 columns]

<IPython.core.display.Javascript object>

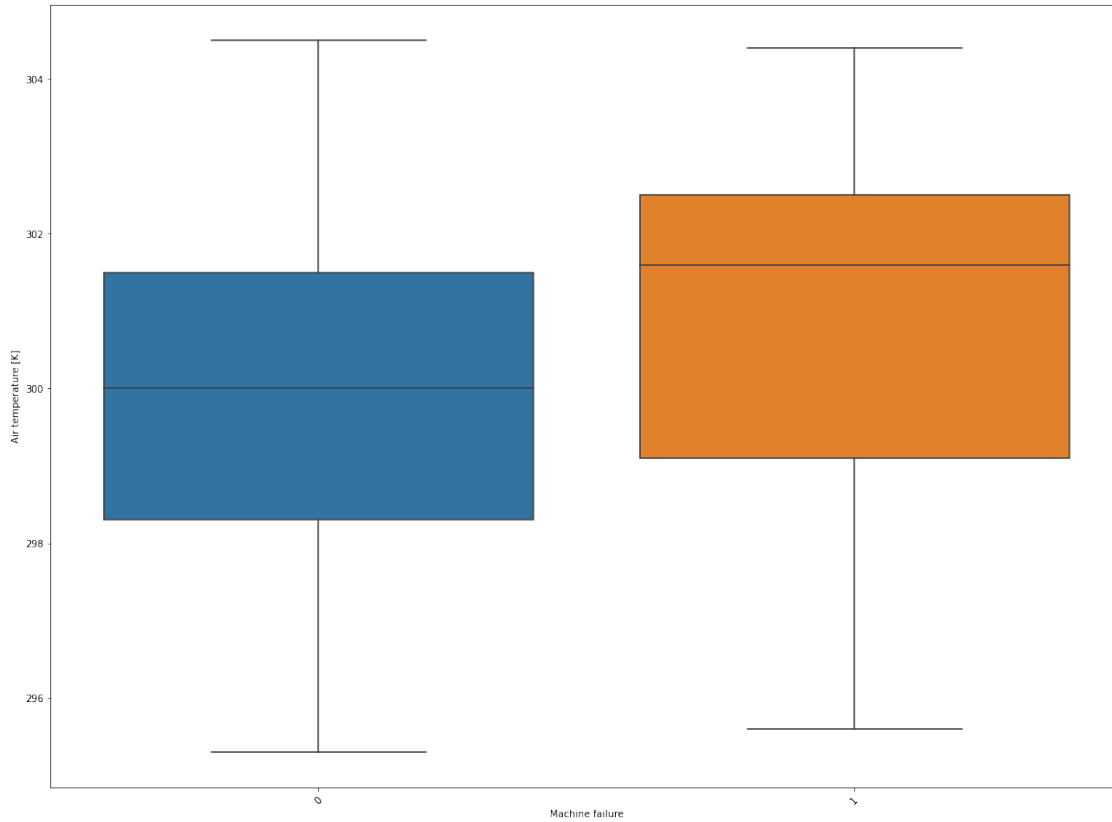
[28]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9be675b390>



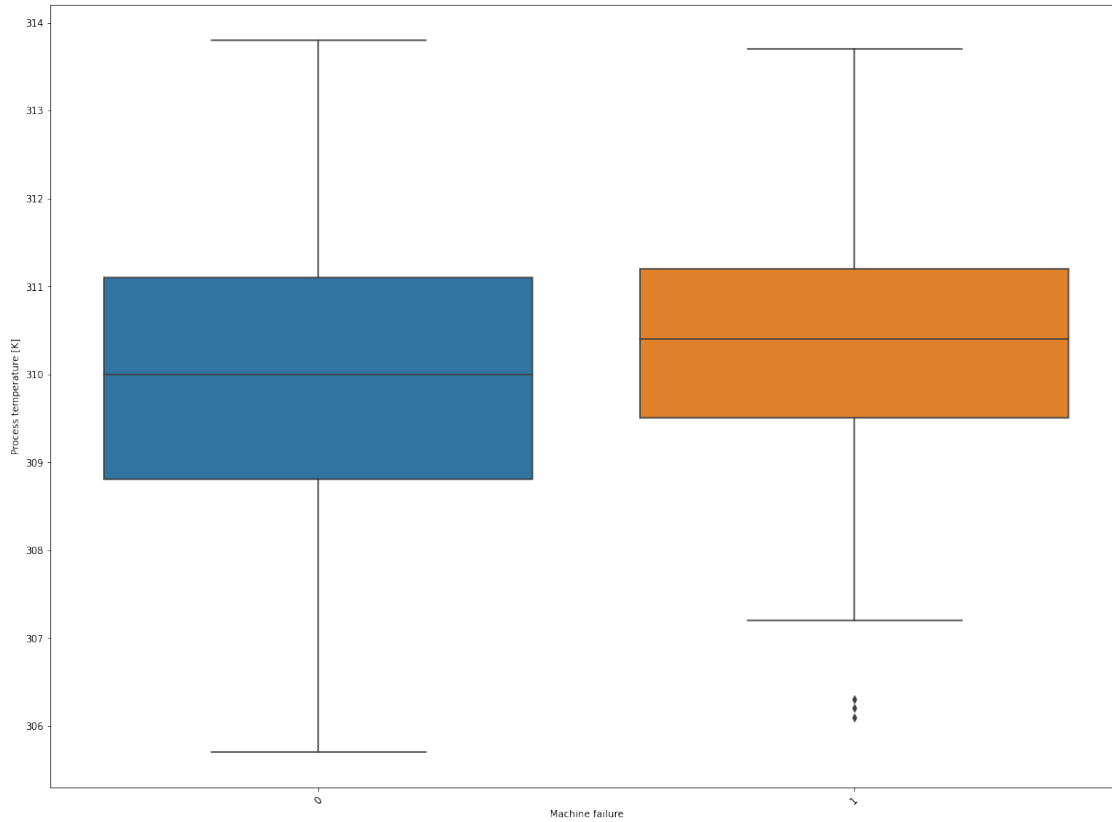
```
[29]: import matplotlib.pyplot as plt
import seaborn as sns
sns.pairplot(df, kind="scatter")
plt.show()
```



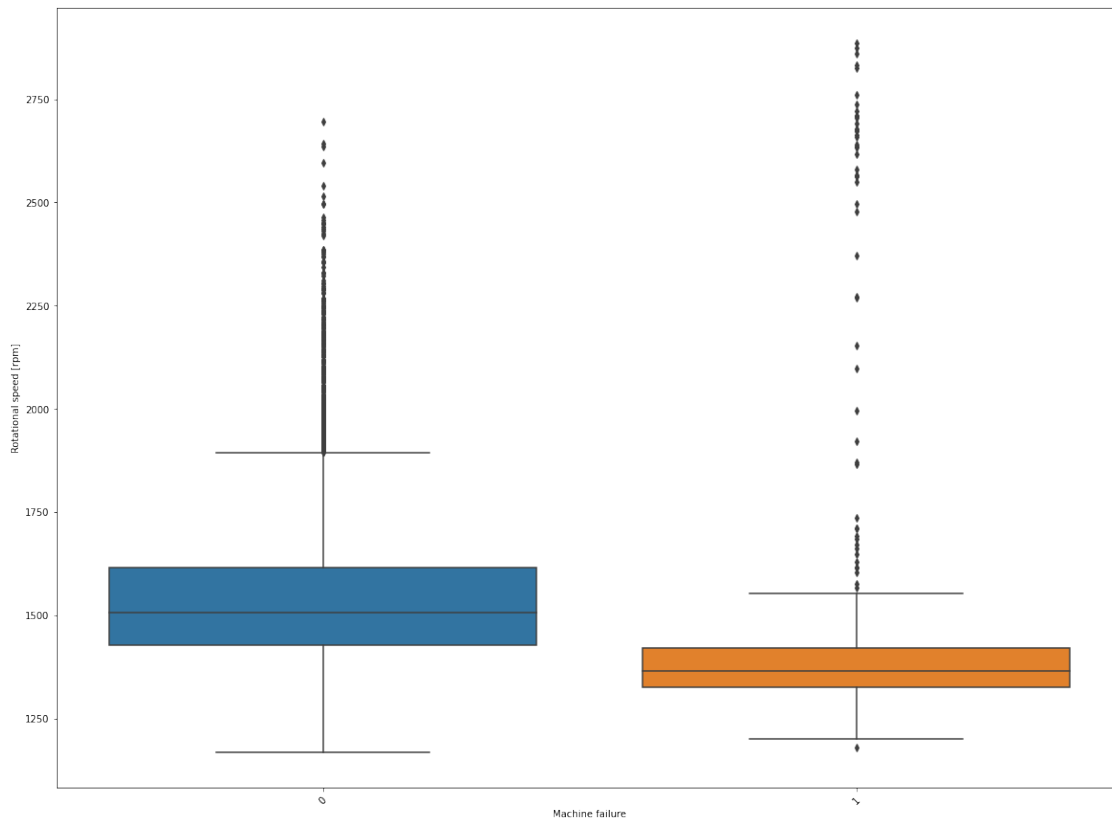
```
[30]: plt.figure(figsize=(20,15))
plt.xticks(rotation=45)
sns.boxplot(data = df, y = 'Air temperature [K]', x = 'Machine failure');
```



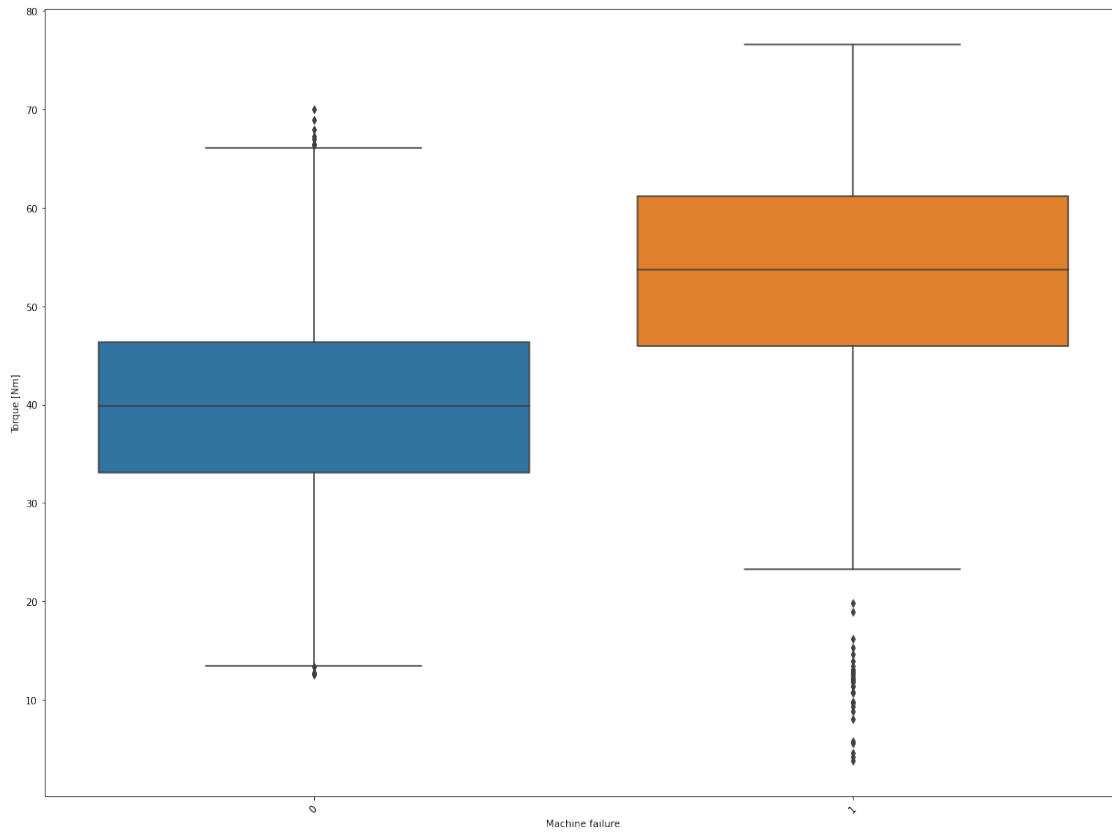
```
[31]: plt.figure(figsize=(20,15))  
plt.xticks(rotation=45)  
sns.boxplot(data = df, y = 'Process temperature [K]', x = 'Machine failure');
```



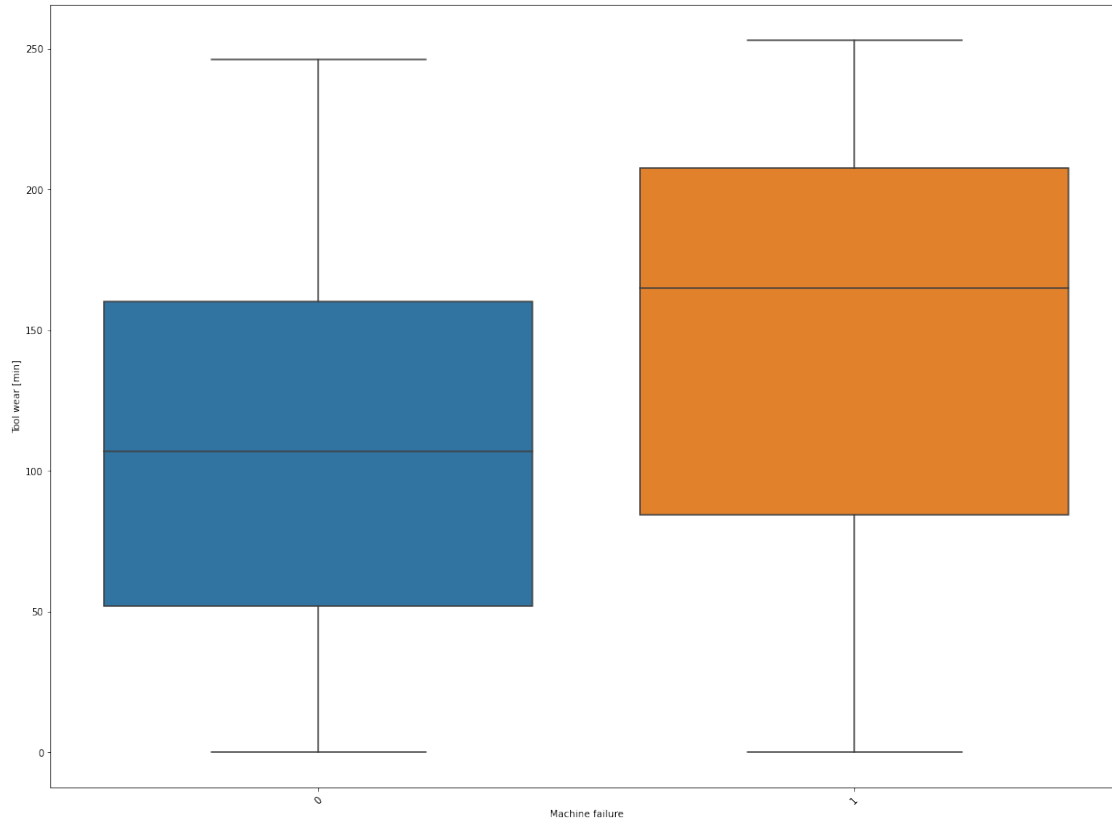
```
[32]: plt.figure(figsize=(20,15))
plt.xticks(rotation=45)
sns.boxplot(data = df, y = 'Rotational speed [rpm]', x = 'Machine failure');
```

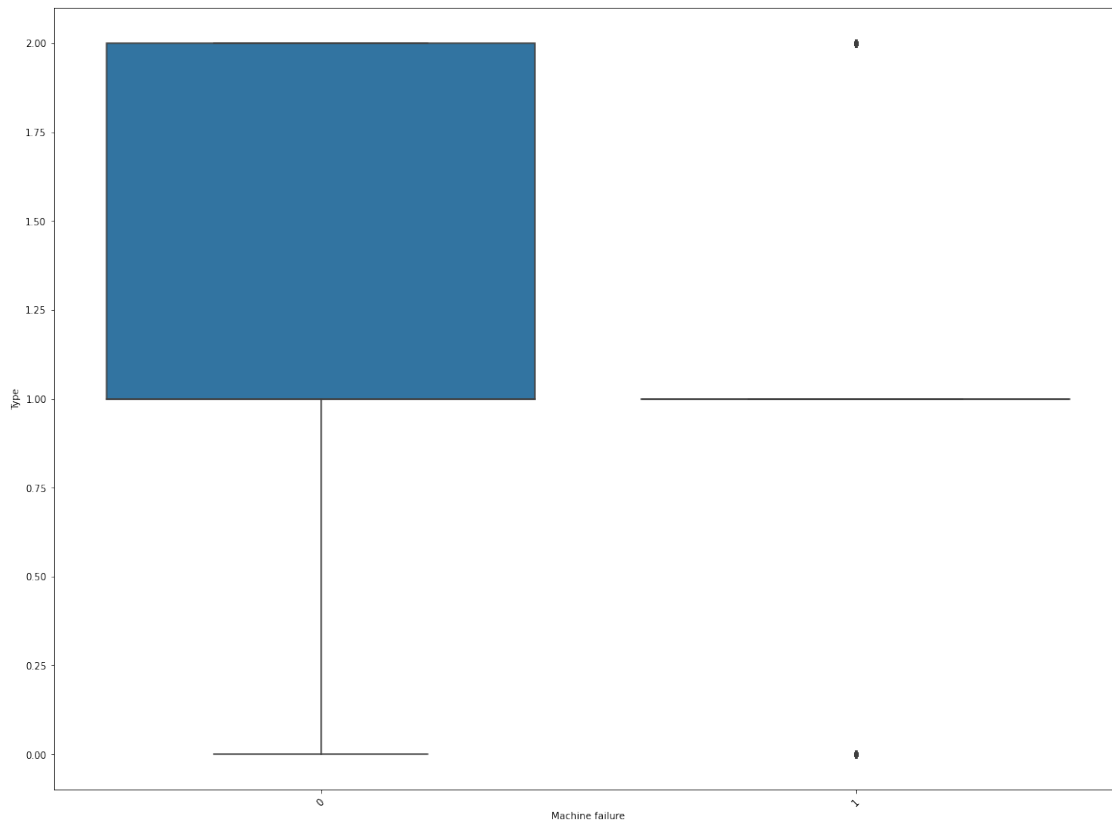
```
[33]: plt.figure(figsize=(20,15))  
plt.xticks(rotation=45)  
sns.boxplot(data = df, y = 'Torque [Nm]', x = 'Machine failure');
```



```
[34]: plt.figure(figsize=(20,15))  
plt.xticks(rotation=45)  
sns.boxplot(data = df, y = 'Tool wear [min]', x = 'Machine failure');
```



```
[35]: plt.figure(figsize=(20,15))  
plt.xticks(rotation=45)  
sns.boxplot(data = df, y = 'Type', x = 'Machine failure');
```



0.0.2 BNN

```
[36]: import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import tensorflow_datasets as tfds
import tensorflow_probability as tfp
```

visualizing data

```
[37]: import matplotlib.pyplot as plt
      %matplotlib inline
      import seaborn as sns

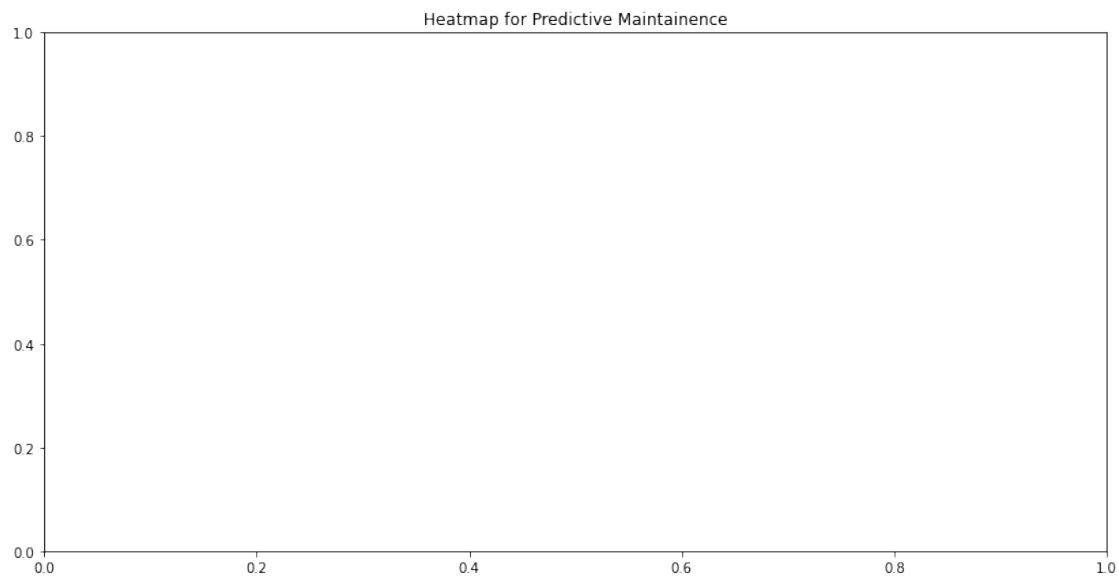
      plt.figure(figsize=(16,6))
      plt.title("Simple Line Plot on Whole Data")
      sns.lineplot(data=df)
```

```
[37]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9bd6e70a90>
```



```
[38]: #heatmaps on whole data
plt.figure(figsize=(14,7))
# Add title
plt.title("Heatmap for Predictive Maintenance")
# Heatmap
#sns.heatmap(data=df['Machine failure'], annot=True)
# Add label for horizontal axis
#plt.xlabel("Axis")
```

```
[38]: Text(0.5, 1.0, 'Heatmap for Predictive Maintenance')
```

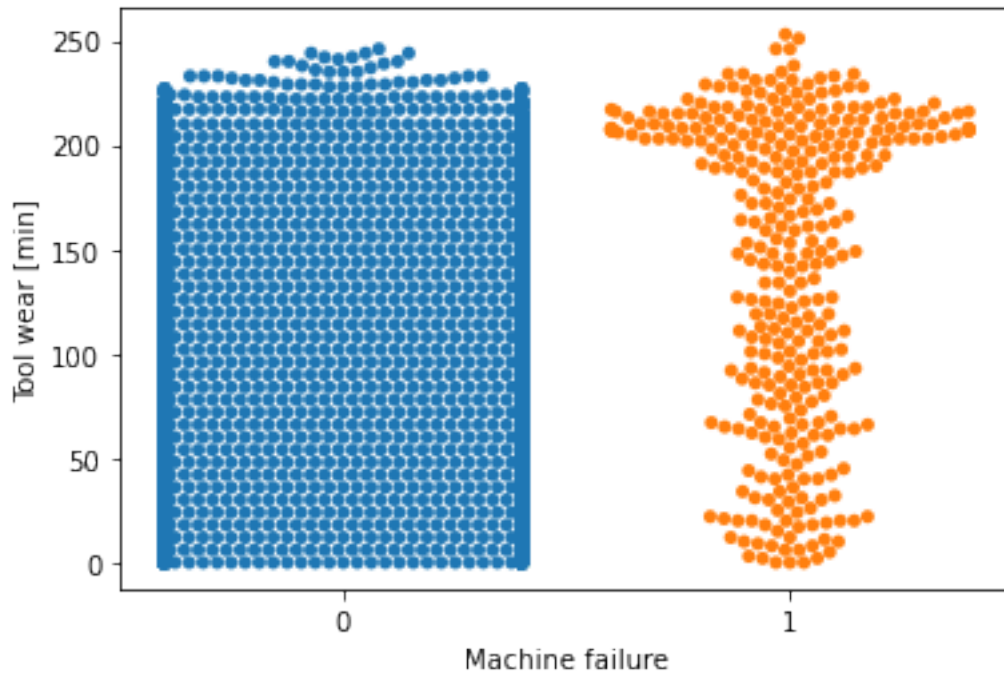


```
[39]: sns.swarmplot(x=df['Machine failure'], y=df['Tool wear [min]'])
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/categorical.py:1296: UserWarning:
89.6% of the points cannot be placed; you may want to decrease the size of the
markers or use stripplot.
```

```
warnings.warn(msg, UserWarning)
```

```
[39]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9b717e7f90>
```

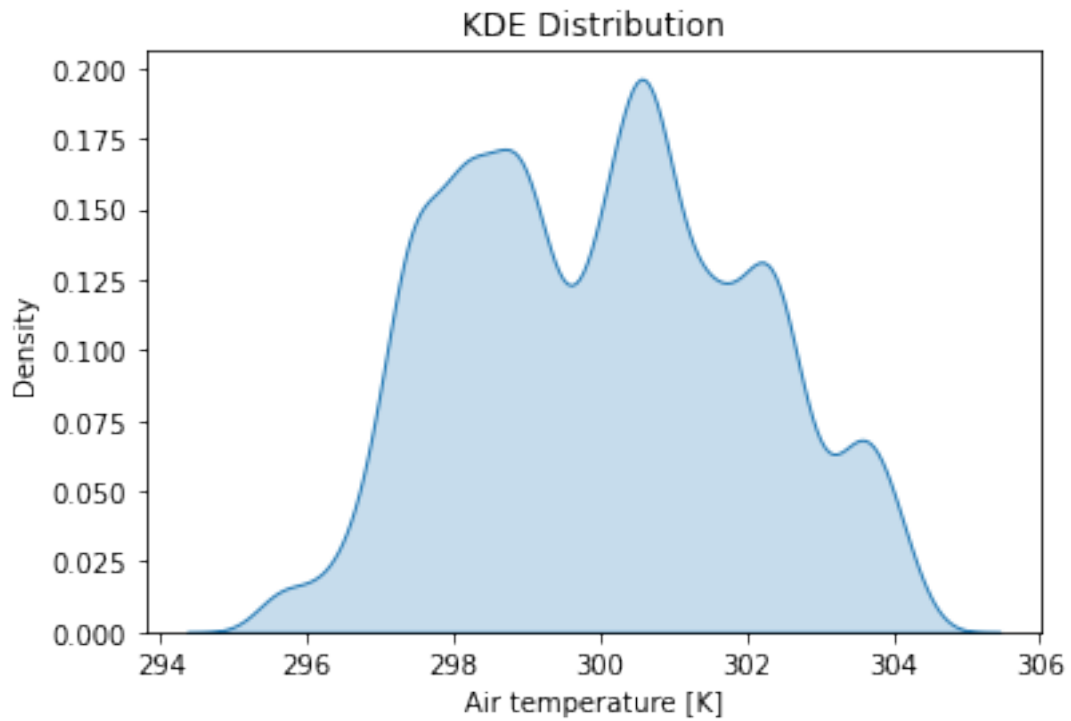


```
[40]: #stripplot
```

```
[41]: #distribution
#for i in df:
sns.kdeplot(data=df['Air temperature [K]'], label='Air temperature [K]',
            shade=True)

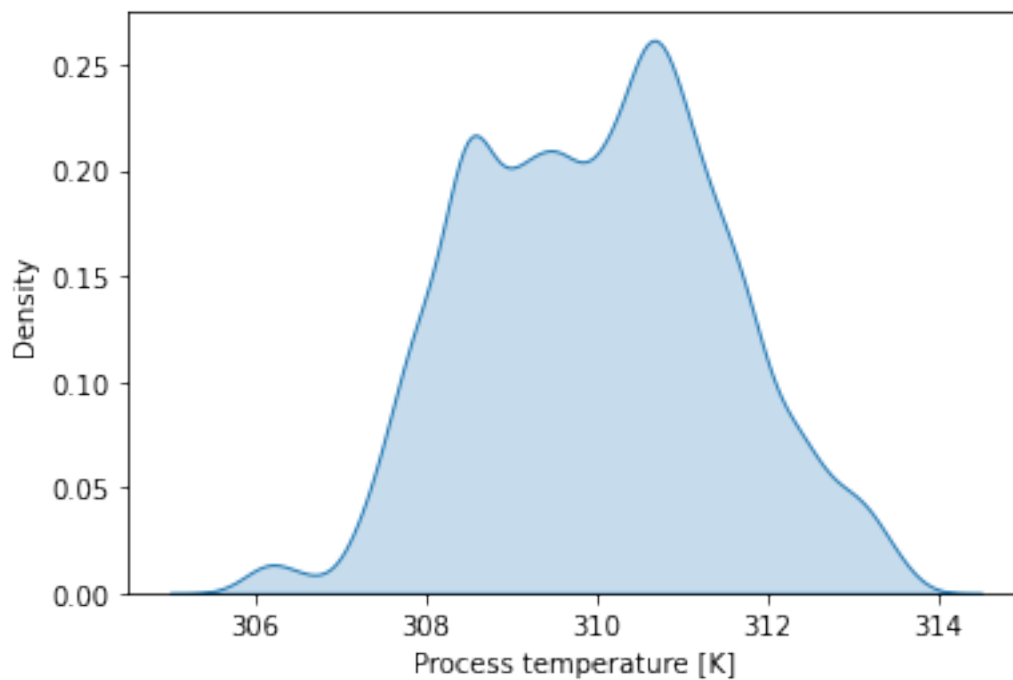
plt.title('KDE Distribution')
```

```
[41]: Text(0.5, 1.0, 'KDE Distribution')
```



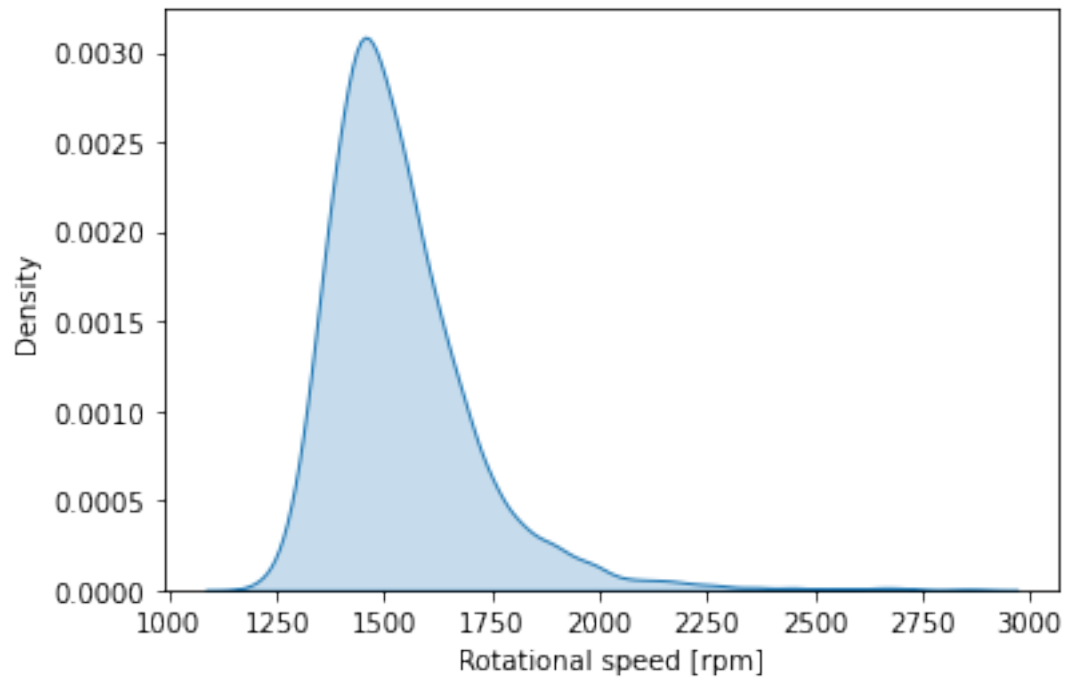
```
[42]: sns.kdeplot(data=df['Process temperature [K]'], label='Process temperature_␣  
→ [K]', shade=True)
```

```
[42]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9b7178c510>
```



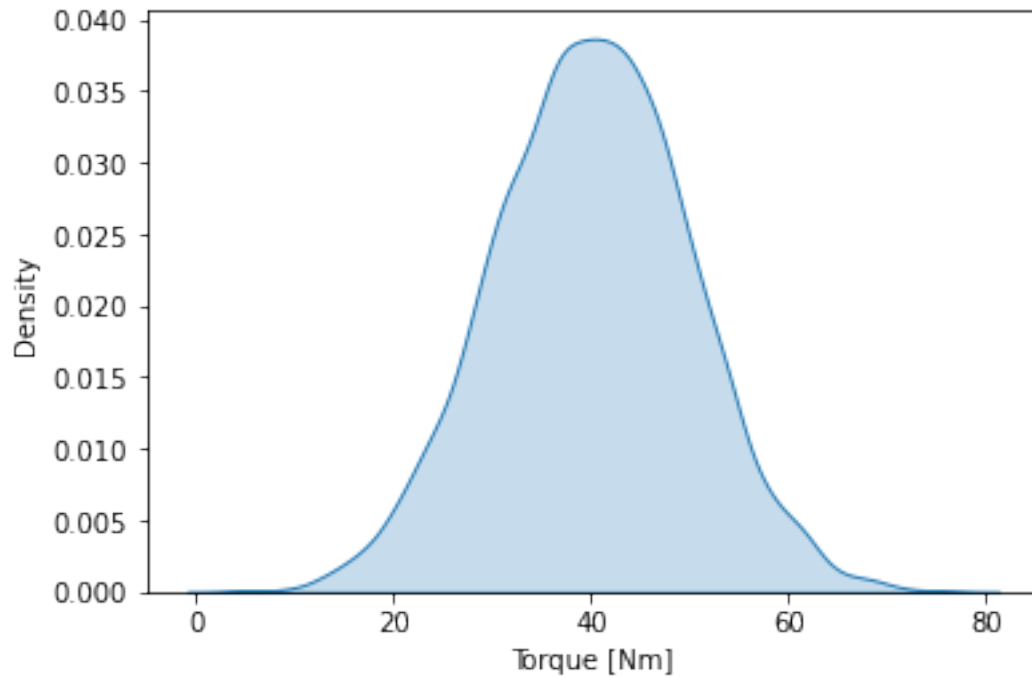
```
[43]: sns.kdeplot(data=df['Rotational speed [rpm]'], label='Rotational speed [rpm]',  
→shade=True)
```

```
[43]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9b716a05d0>
```



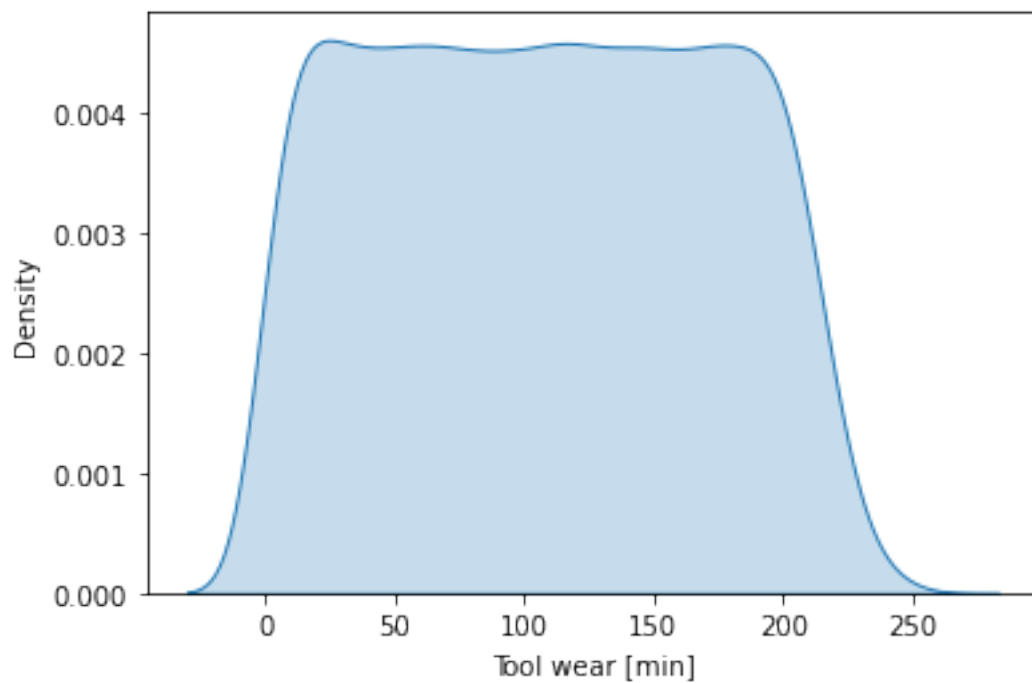
```
[44]: sns.kdeplot(data=df['Torque [Nm]'], label='Torque [Nm]', shade=True)
```

```
[44]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9b71661610>
```

```
[45]: sns.kdeplot(data=df['Tool wear [min]'], label='Tool wear [min]', shade=True)
```

```
[45]: <matplotlib.axes._subplots.AxesSubplot at 0x7f9b6f4e5490>
```



Create training and evaluation datasets

```
[46]: df.columns
```

```
[46]: Index(['Type', 'Air temperature [K]', 'Process temperature [K]',  
        'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]',  
        'Machine failure', 'TWF', 'HDF', 'PWF', 'OSF', 'RNF'],  
        dtype='object')
```

```
[47]: from sklearn.model_selection import train_test_split  
  
#first moving target variable "Machine Failure" to end and then defining X and  
→y  
df = df[['Type', 'Air temperature [K]', 'Process temperature [K]',  
        'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]',  
        'TWF', 'HDF', 'PWF', 'OSF', 'RNF', 'Machine failure']]  
print(df.shape)  
# excluding last variable for target variable  
X = df.iloc[:, :-1]  
print(X.shape)  
# making last variable as target variable  
y = df.iloc[:, -1]  
print(y.shape)  
# using 70:30 split for making training and testing datasets and using random  
→state as 42 to repeat this random split.  
X_train,X_test,y_train,y_test = train_test_split(X, y,test_size=0.  
→3,random_state=42)
```

```
(10000, 12)  
(10000, 11)  
(10000,)
```

```
[48]: # the shapes of X_train,X_test,y_train,y_test  
print(X_train.shape)  
print(X_test.shape)  
print(y_train.shape)  
print(y_test.shape)
```

```
(7000, 11)  
(3000, 11)  
(7000,)  
(3000,)
```

```
[49]: print(X_train.shape)  
print(y_train.shape)
```

```
(7000, 11)  
(7000,)
```

```
[50]: y_train.head()
```

```
[50]: 9069    0
      2603    0
      7738    0
      1579    0
      5058    0
      Name: Machine failure, dtype: int64
```

```
[51]: # correct
      #done
      #train dataset
      train_d = pd.DataFrame(X_train)
      train_d['y_train'] = y_train
      print(train_d.shape)
      print(train_d.shape)

      #test dataset
      test_d = pd.DataFrame(X_test)
      test_d['y_test'] = y_test
      print(test_d.shape)
      print(test_d.shape)
```

```
(7000, 12)
```

```
(7000, 12)
```

```
(3000, 12)
```

```
(3000, 12)
```

```
[52]: train_d.head()
```

```
[52]:
```

	Type	Air temperature [K]	Process temperature [K]	...	OSF	RNF	y_train
9069	2.0	297.2	308.2	...	0	0	0
2603	2.0	299.3	309.2	...	0	0	0
7738	2.0	300.5	312.0	...	0	0	0
1579	1.0	298.3	308.3	...	0	0	0
5058	1.0	303.9	312.9	...	0	0	0

```
[5 rows x 12 columns]
```

```
[53]: test_d.head()
```

```
[53]:
```

	Type	Air temperature [K]	Process temperature [K]	...	OSF	RNF	y_test
6252	1.0	300.8	310.3	...	0	0	0
4684	2.0	303.6	311.8	...	0	0	1
1731	2.0	298.3	307.9	...	0	0	0
4742	1.0	303.3	311.3	...	0	0	0
4521	1.0	302.4	310.4	...	0	0	0

```
[5 rows x 12 columns]
```

Compile, train, and evaluate the model

[54]: *# from here will write in the form of functions*

Create model inputs

Experiment 1: standard neural network

```
[55]: from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import cross_val_score
from keras.models import Sequential # to initialize NN
from keras.layers import Dense # to build layers

classifier = Sequential()
classifier.add(Dense(units = 5, input_dim = X_train.shape[1])) # changed this
classifier.add(Dense(units = 3, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = _
    → ['accuracy'])
classifier.fit(X_train, y_train, epochs=50)
test_loss, test_acc = classifier.evaluate(X_test, y_test, verbose=2)
print('\nTest accuracy:', test_acc)
print('\nTest loss:', test_loss)
```

Epoch 1/50

219/219 [=====] - 1s 2ms/step - loss: 7.3479 -
accuracy: 0.8310

Epoch 2/50

219/219 [=====] - 0s 2ms/step - loss: 1.0049 -
accuracy: 0.9649

Epoch 3/50

219/219 [=====] - 0s 2ms/step - loss: 0.5311 -
accuracy: 0.9649

Epoch 4/50

219/219 [=====] - 0s 1ms/step - loss: 0.4266 -
accuracy: 0.9649

Epoch 5/50

219/219 [=====] - 0s 2ms/step - loss: 0.3700 -
accuracy: 0.9649

Epoch 6/50

219/219 [=====] - 0s 2ms/step - loss: 0.3274 -
accuracy: 0.9649

Epoch 7/50

219/219 [=====] - 0s 2ms/step - loss: 0.2941 -
accuracy: 0.9649

Epoch 8/50

219/219 [=====] - 0s 2ms/step - loss: 0.2676 -
accuracy: 0.9649

Epoch 9/50
219/219 [=====] - 0s 1ms/step - loss: 0.2462 -
accuracy: 0.9649
Epoch 10/50
219/219 [=====] - 0s 2ms/step - loss: 0.2289 -
accuracy: 0.9649
Epoch 11/50
219/219 [=====] - 0s 1ms/step - loss: 0.2148 -
accuracy: 0.9649
Epoch 12/50
219/219 [=====] - 0s 1ms/step - loss: 0.2032 -
accuracy: 0.9649
Epoch 13/50
219/219 [=====] - 0s 2ms/step - loss: 0.1936 -
accuracy: 0.9649
Epoch 14/50
219/219 [=====] - 0s 1ms/step - loss: 0.1858 -
accuracy: 0.9649
Epoch 15/50
219/219 [=====] - 0s 1ms/step - loss: 0.1793 -
accuracy: 0.9649
Epoch 16/50
219/219 [=====] - 0s 1ms/step - loss: 0.1739 -
accuracy: 0.9649
Epoch 17/50
219/219 [=====] - 0s 2ms/step - loss: 0.1696 -
accuracy: 0.9649
Epoch 18/50
219/219 [=====] - 0s 2ms/step - loss: 0.1660 -
accuracy: 0.9649
Epoch 19/50
219/219 [=====] - 0s 2ms/step - loss: 0.1631 -
accuracy: 0.9649
Epoch 20/50
219/219 [=====] - 0s 2ms/step - loss: 0.1607 -
accuracy: 0.9649
Epoch 21/50
219/219 [=====] - 0s 1ms/step - loss: 0.1588 -
accuracy: 0.9649
Epoch 22/50
219/219 [=====] - 0s 2ms/step - loss: 0.1572 -
accuracy: 0.9649
Epoch 23/50
219/219 [=====] - 0s 1ms/step - loss: 0.1560 -
accuracy: 0.9649
Epoch 24/50
219/219 [=====] - 0s 2ms/step - loss: 0.1550 -
accuracy: 0.9649

Epoch 25/50
219/219 [=====] - 0s 2ms/step - loss: 0.1543 -
accuracy: 0.9649
Epoch 26/50
219/219 [=====] - 0s 2ms/step - loss: 0.1537 -
accuracy: 0.9649
Epoch 27/50
219/219 [=====] - 0s 1ms/step - loss: 0.1533 -
accuracy: 0.9649
Epoch 28/50
219/219 [=====] - 0s 2ms/step - loss: 0.1530 -
accuracy: 0.9649
Epoch 29/50
219/219 [=====] - 0s 1ms/step - loss: 0.1528 -
accuracy: 0.9649
Epoch 30/50
219/219 [=====] - 0s 1ms/step - loss: 0.1526 -
accuracy: 0.9649
Epoch 31/50
219/219 [=====] - 0s 2ms/step - loss: 0.1525 -
accuracy: 0.9649
Epoch 32/50
219/219 [=====] - 0s 1ms/step - loss: 0.1524 -
accuracy: 0.9649
Epoch 33/50
219/219 [=====] - 0s 2ms/step - loss: 0.1523 -
accuracy: 0.9649
Epoch 34/50
219/219 [=====] - 0s 2ms/step - loss: 0.1523 -
accuracy: 0.9649
Epoch 35/50
219/219 [=====] - 0s 2ms/step - loss: 0.1523 -
accuracy: 0.9649
Epoch 36/50
219/219 [=====] - 0s 1ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 37/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 38/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 39/50
219/219 [=====] - 0s 1ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 40/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649

```

Epoch 41/50
219/219 [=====] - 0s 1ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 42/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 43/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 44/50
219/219 [=====] - 0s 1ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 45/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 46/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 47/50
219/219 [=====] - 0s 1ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 48/50
219/219 [=====] - 0s 1ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 49/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
Epoch 50/50
219/219 [=====] - 0s 2ms/step - loss: 0.1522 -
accuracy: 0.9649
94/94 - 0s - loss: 0.1385 - accuracy: 0.9690 - 226ms/epoch - 2ms/step

```

Test accuracy: 0.968999981880188

Test loss: 0.13847528398036957

train accuracy: 0.9649, loss: 0.1522 after 50 epochs test accuracy: 0.9690, loss: 0.1385

[55]:

[56]:

```
classifier.summary()
```

Model: "sequential"

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 5)	65
dense_1 (Dense)	(None, 3)	18

dense_2 (Dense) (None, 1) 4

```
=====
Total params: 87
Trainable params: 87
Non-trainable params: 0
-----
```

```
[57]: # checking the probabilities
probability_model = Sequential([classifier, tf.keras.layers.Softmax()])
predictions = probability_model.predict(X_test)
predictions[0]
```

```
[57]: array([1.], dtype=float32)
```

```
[58]: np.argmax(predictions[0])
```

```
[58]: 0
```

```
[59]: y_test[0]
```

```
[59]: 0
```

```
[60]: predictions
```

```
[60]: array([[1.],
          [1.],
          [1.],
          ...,
          [1.],
          [1.],
          [1.]], dtype=float32)
```

```
[61]: y_test.nunique
```

```
[61]: <bound method IndexOpsMixin.nunique of 6252 0
4684    1
1731    0
4742    0
4521    0
..
8014    0
1074    0
3063    0
6487    0
4705    0
Name: Machine failure, Length: 3000, dtype: int64>
```

Experiment 2: Bayesian neural network (BNN)

dependencies and prerequisites

```
[62]: from pprint import pprint
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns

import tensorflow.compat.v2 as tf
tf.enable_v2_behavior()

import tensorflow_probability as tfp

sns.reset_defaults()
sns.set_context(context='talk', font_scale=0.7)
plt.rcParams['image.cmap'] = 'viridis'

%matplotlib inline

tfd = tfp.distributions
tfb = tfp.bijectors
```

define priors and other functions

```
[63]: # to build the bnn
```

define bnn functions and class

```
[64]: from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import cross_val_score
from keras.models import Sequential # to initialize NN
from keras.layers import Dense # to build layers
'''
classifier = Sequential()
classifier.add(Dense(units = 8, input_dim = X_train.shape[1])) # changed this
classifier.add(Dense(units = 4, activation = 'relu'))
classifier.add(Dense(units = 1, activation = 'sigmoid'))
classifier.compile(optimizer = 'adam', loss = 'binary_crossentropy', metrics = ['accuracy'])
classifier.fit(X_train, y_train, epochs=100)
test_loss, test_acc = classifier.evaluate(X_test, y_test, verbose=2)
print('\nTest accuracy:', test_acc)
'''

[64]: "\nclassifier = Sequential()\nclassifier.add(Dense(units = 8, input_dim =
X_train.shape[1])) # changed this\nclassifier.add(Dense(units = 4, activation =
'relu'))\nclassifier.add(Dense(units = 1, activation =
'sigmoid'))\nclassifier.compile(optimizer = 'adam', loss =
'binary_crossentropy', metrics = ['accuracy'])\nclassifier.fit(X_train, y_train,
```

```
epochs=100)\ntest_loss, test_acc = classifier.evaluate(X_test, y_test,
verbose=2)\nprint('\nTest accuracy:', test_acc)\n\n"
```

target is machine failure variable

```
[65]: from sklearn.model_selection import train_test_split

#first moving target variable "Machine Failure" to end and then defining X and
→y
df = df[['Type', 'Air temperature [K]', 'Process temperature [K]',
        'Rotational speed [rpm]', 'Torque [Nm]', 'Tool wear [min]',
        'TWF', 'HDF', 'PWF', 'OSF', 'RNF', 'Machine failure']]
print(df.shape)
# excluding last variable for target variable
X = df.iloc[:, :-1]
print(X.shape)
# making last variable as target variable
y = df.iloc[:, -1]
print(y.shape)
# using 70:30 split for making training and testing datasets and using random
→state as 42 to repeat this random split.
X_train,X_test,y_train,y_test = train_test_split(X, y,test_size=0.
→3,random_state=42)
```

```
(10000, 12)
```

```
(10000, 11)
```

```
(10000,)
```

```
[66]: dist = tfp.distributions
dataset_size = len(X_train)
kl_divergence_function = (lambda q, p, _: dist.kl_divergence(q, p) / tf.
→cast(dataset_size, dtype=tf.float32))

model_tfp = tf.keras.Sequential([
    tf.keras.Input(X_train.shape[1]),
    tfp.layers.DenseFlipout(16, kernel_divergence_fn=kl_divergence_function,
→activation=tf.nn.relu),
    tfp.layers.DenseFlipout(6, kernel_divergence_fn=kl_divergence_function,
→activation=tf.nn.relu ),
    tfp.layers.DenseFlipout(3, kernel_divergence_fn=kl_divergence_function,
→activation=tf.nn.softmax),
])

learning_rate = 0.001
model_tfp.compile(optimizer=tf.keras.optimizers.
→Adam(learning_rate),loss='binary_crossentropy',metrics=['accuracy'])
```

```
/usr/local/lib/python3.7/dist-
```

```

packages/tensorflow_probability/python/layers/util.py:102: UserWarning:
`layer.add_variable` is deprecated and will be removed in a future version.
Please use `layer.add_weight` method instead.
    trainable=trainable)
/usr/local/lib/python3.7/dist-
packages/tensorflow_probability/python/layers/util.py:112: UserWarning:
`layer.add_variable` is deprecated and will be removed in a future version.
Please use `layer.add_weight` method instead.
    trainable=trainable)

```

```

[67]: model_tfp.fit(X_train, y_train, epochs=50)
      test_loss, test_acc = model_tfp.evaluate(X_test, y_test)
      print('\nTest accuracy:', test_acc)
      print('\nTest loss:', test_loss)

```

```

Epoch 1/50
219/219 [=====] - 2s 2ms/step - loss: 0.7325 -
accuracy: 0.1427
Epoch 2/50
219/219 [=====] - 0s 2ms/step - loss: 0.5319 -
accuracy: 0.0171
Epoch 3/50
219/219 [=====] - 1s 2ms/step - loss: 0.5255 -
accuracy: 0.0069
Epoch 4/50
219/219 [=====] - 1s 2ms/step - loss: 0.5212 -
accuracy: 0.0034
Epoch 5/50
219/219 [=====] - 1s 2ms/step - loss: 0.5166 -
accuracy: 0.6636
Epoch 6/50
219/219 [=====] - 1s 2ms/step - loss: 0.5122 -
accuracy: 0.6729
Epoch 7/50
219/219 [=====] - 0s 2ms/step - loss: 0.5082 -
accuracy: 0.3919
Epoch 8/50
219/219 [=====] - 0s 2ms/step - loss: 0.5044 -
accuracy: 0.2486
Epoch 9/50
219/219 [=====] - 0s 2ms/step - loss: 0.5004 -
accuracy: 0.4834
Epoch 10/50
219/219 [=====] - 0s 2ms/step - loss: 0.4968 -
accuracy: 0.1660
Epoch 11/50
219/219 [=====] - 0s 2ms/step - loss: 0.4934 -

```

```

accuracy: 0.0294
Epoch 12/50
219/219 [=====] - 0s 2ms/step - loss: 0.4903 -
accuracy: 0.0201
Epoch 13/50
219/219 [=====] - 0s 2ms/step - loss: 0.4874 -
accuracy: 0.0141
Epoch 14/50
219/219 [=====] - 0s 2ms/step - loss: 0.4858 -
accuracy: 0.0207
Epoch 15/50
219/219 [=====] - 0s 2ms/step - loss: 0.4824 -
accuracy: 0.4201
Epoch 16/50
219/219 [=====] - 1s 2ms/step - loss: 0.4805 -
accuracy: 0.0941
Epoch 17/50
219/219 [=====] - 0s 2ms/step - loss: 0.4807 -
accuracy: 0.4541
Epoch 18/50
219/219 [=====] - 0s 2ms/step - loss: 0.4769 -
accuracy: 0.1937
Epoch 19/50
219/219 [=====] - 0s 2ms/step - loss: 0.4756 -
accuracy: 0.4201
Epoch 20/50
219/219 [=====] - 1s 2ms/step - loss: 0.4739 -
accuracy: 0.3471
Epoch 21/50
219/219 [=====] - 0s 2ms/step - loss: 0.4727 -
accuracy: 0.4606
Epoch 22/50
219/219 [=====] - 0s 2ms/step - loss: 0.4713 -
accuracy: 0.6354
Epoch 23/50
219/219 [=====] - 0s 2ms/step - loss: 0.4701 -
accuracy: 0.1963
Epoch 24/50
219/219 [=====] - 0s 2ms/step - loss: 0.4703 -
accuracy: 0.4379
Epoch 25/50
219/219 [=====] - 0s 2ms/step - loss: 0.4678 -
accuracy: 0.4039
Epoch 26/50
219/219 [=====] - 0s 2ms/step - loss: 0.4667 -
accuracy: 0.2937
Epoch 27/50
219/219 [=====] - 0s 2ms/step - loss: 0.4661 -

```

```

accuracy: 0.3123
Epoch 28/50
219/219 [=====] - 0s 2ms/step - loss: 0.4648 -
accuracy: 0.1520
Epoch 29/50
219/219 [=====] - 0s 2ms/step - loss: 0.4649 -
accuracy: 0.3357
Epoch 30/50
219/219 [=====] - 0s 2ms/step - loss: 0.4632 -
accuracy: 0.3991
Epoch 31/50
219/219 [=====] - 0s 2ms/step - loss: 0.4627 -
accuracy: 0.2421
Epoch 32/50
219/219 [=====] - 0s 2ms/step - loss: 0.4617 -
accuracy: 0.3357
Epoch 33/50
219/219 [=====] - 0s 2ms/step - loss: 0.4610 -
accuracy: 0.1270
Epoch 34/50
219/219 [=====] - 0s 2ms/step - loss: 0.4601 -
accuracy: 0.4441
Epoch 35/50
219/219 [=====] - 0s 2ms/step - loss: 0.4605 -
accuracy: 0.9087
Epoch 36/50
219/219 [=====] - 0s 2ms/step - loss: 0.4586 -
accuracy: 0.3583
Epoch 37/50
219/219 [=====] - 0s 2ms/step - loss: 0.4594 -
accuracy: 0.2394
Epoch 38/50
219/219 [=====] - 0s 2ms/step - loss: 0.4572 -
accuracy: 0.3223
Epoch 39/50
219/219 [=====] - 0s 2ms/step - loss: 0.4576 -
accuracy: 0.5739
Epoch 40/50
219/219 [=====] - 1s 2ms/step - loss: 0.4567 -
accuracy: 0.5314
Epoch 41/50
219/219 [=====] - 0s 2ms/step - loss: 0.4566 -
accuracy: 0.3981
Epoch 42/50
219/219 [=====] - 0s 2ms/step - loss: 0.4549 -
accuracy: 0.4343
Epoch 43/50
219/219 [=====] - 0s 2ms/step - loss: 0.4559 -

```

```

accuracy: 0.6851
Epoch 44/50
219/219 [=====] - 0s 2ms/step - loss: 0.4546 -
accuracy: 0.4286
Epoch 45/50
219/219 [=====] - 0s 2ms/step - loss: 0.4541 -
accuracy: 0.2433
Epoch 46/50
219/219 [=====] - 1s 2ms/step - loss: 0.4529 -
accuracy: 0.4510
Epoch 47/50
219/219 [=====] - 1s 2ms/step - loss: 0.4538 -
accuracy: 0.9517
Epoch 48/50
219/219 [=====] - 0s 2ms/step - loss: 0.4540 -
accuracy: 0.4907
Epoch 49/50
219/219 [=====] - 1s 2ms/step - loss: 0.4518 -
accuracy: 0.5821
Epoch 50/50
219/219 [=====] - 0s 2ms/step - loss: 0.4527 -
accuracy: 0.2293
94/94 [=====] - 1s 2ms/step - loss: 0.4495 - accuracy:
0.9687

```

Test accuracy: 0.968666672706604

Test loss: 0.44948306679725647

Test accuracy: 0.968999981880188 after 50 epochs and test loss:

[68]: `model_tfp.summary()`

Model: "sequential_2"

Layer (type)	Output Shape	Param #
dense_flipout (DenseFlipout)	(None, 16)	368
dense_flipout_1 (DenseFlipout)	(None, 6)	198
dense_flipout_2 (DenseFlipout)	(None, 3)	39

Total params: 605

Trainable params: 605

Non-trainable params: 0

define tensorboard variables for we plots

Train BNN with a small training subset.

Train BNN with the whole training set.

Experiment 3: probabilistic Bayesian neural network w and b site streamlit for gui

[68]:

0.0.3 WEEKLY OUTPUT PDFS

convert notebook to pdf for weekly progrss submission

[68]:

[69]: %cd /content/drive/MyDrive/Colab Notebooks/MTP

!pwd

!ls

```
/content/drive/MyDrive/Colab Notebooks/MTP
/content/drive/MyDrive/Colab Notebooks/MTP
3rd_sem1.pdf    material          READ.md         'web app'
3rd_sem.pdf     MTP_BNN.ipynb    README.md
datasets       MTP_BNN.pdf      w1.pdf
dec.pdf        MTP_Data_Visualization.ipynb  w2.pdf
```

```
[ ]: !sudo apt-get install texlive-xetex texlive-fonts-recommended
      ↳texlive-generic-recommended
```

Reading package lists... Done

Building dependency tree

Reading state information... Done

The following additional packages will be installed:

fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
libruby2.5 libsyntaxtex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
rubygems-integration tlutils tex-common tex-gyre texlive-base
texlive-binaries texlive-latex-base texlive-latex-extra
texlive-latex-recommended texlive-pictures texlive-plain-generic tipa

Suggested packages:

fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript

```

fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
| fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
| pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
python-pygments icc-profiles libfile-which-perl
libspreadsheet-parseexcel-perl texlive-latex-extra-doc
texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
| libtcltk-ruby texlive-pictures-doc vprerex

```

The following NEW packages will be installed:

```

fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
libruby2.5 libsynchronet1 libtexlua52 libtexluajit2 libzip-0-13 lmodern
poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
rubygems-integration tclutils tex-common tex-gyre texlive-base
texlive-binaries texlive-fonts-recommended texlive-generic-recommended
texlive-latex-base texlive-latex-extra texlive-latex-recommended
texlive-pictures texlive-plain-generic texlive-xetex tipa

```

0 upgraded, 47 newly installed, 0 to remove and 37 not upgraded.

Need to get 146 MB of archives.

After this operation, 460 MB of additional disk space will be used.

Get:1 <http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback>
all 1:6.0.1r16-1.1 [1,805 kB]

Get:2 <http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato> all 2.0-2
[2,698 kB]

Get:3 <http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data> all
0.4.8-2 [1,479 kB]

Get:4 <http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common> all 6.09
[33.0 kB]

Get:5 <http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern> all
2.004.5-3 [4,551 kB]

Get:6 <http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono> all
20171026-2 [75.5 kB]

Get:7 <http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre> all
20160520-1 [8,761 kB]

Get:8 <http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common> all
11 [6,066 B]

Get:9 <http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1>
amd64 1.20.2-0ubuntu3.1 [108 kB]

Get:10 <http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2>
amd64 2.2.7-1ubuntu2.8 [18.6 kB]

Get:11 <http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35> amd64
0.35-13 [15.5 kB]

Get:12 <http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0> amd64
0.13-6 [55.9 kB]

Get:13 <http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common>
all 9.26~dfsg+0-0ubuntu0.18.04.14 [5,092 kB]

Get:14 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libgs9 amd64 9.26~dfsg+0-0ubuntu0.18.04.14 [2,265 kB]
Get:15 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 libjs-jquery all 3.2.1-1 [152 kB]
Get:16 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libkpathsea6 amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 libpotrace0 amd64 1.14-2 [17.4 kB]
Get:18 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libptexenc1 amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 rubygems-integration all 1.11 [4,994 B]
Get:20 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 ruby2.5 amd64 2.5.1-1ubuntu1.10 [48.6 kB]
Get:21 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby amd64 1:2.5.1 [5,712 B]
Get:22 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 rake all 12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-did-you-mean all 1.2.0-2 [9,700 B]
Get:24 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-minitest all 5.10.3-1 [38.6 kB]
Get:25 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-net-telnet all 0.1.1-2 [12.6 kB]
Get:26 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-power-assert all 0.3.0-1 [7,952 B]
Get:27 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 ruby-test-unit all 3.2.5-1 [61.1 kB]
Get:28 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libruby2.5 amd64 2.5.1-1ubuntu1.10 [3,071 kB]
Get:29 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libsyntax1 amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libtexlua52 amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libtexluajit2 amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 libzip-0-13 amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 lmodern all 2.004.5-3 [9,631 kB]
Get:34 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 preview-latex-style all 11.91-1ubuntu1 [185 kB]
Get:35 <http://archive.ubuntu.com/ubuntu> bionic/main amd64 t1utils amd64 1.41-2 [56.0 kB]
Get:36 <http://archive.ubuntu.com/ubuntu> bionic/universe amd64 tex-gyre all 20160520-1 [4,998 kB]
Get:37 <http://archive.ubuntu.com/ubuntu> bionic-updates/main amd64 texlive-binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]

```

Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-generic-
recommended all 2017.20180305-1 [15.9 kB]
Get:42 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
2017.20180305-1 [951 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 8s (17.5 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 76,
<> line 47.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 155222 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb ...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2_all.deb ...
Unpacking fonts-lato (2.0-2) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.8-2_all.deb ...
Unpacking poppler-data (0.4.8-2) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.09_all.deb ...
Unpacking tex-common (6.09) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../04-fonts-lmodern_2.004.5-3_all.deb ...
Unpacking fonts-lmodern (2.004.5-3) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../05-fonts-noto-mono_20171026-2_all.deb ...

```

```

Unpacking fonts-noto-mono (20171026-2) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../06-fonts-texgyre_20160520-1_all.deb ...
Unpacking fonts-texgyre (20160520-1) ...
Selecting previously unselected package javascript-common.
Preparing to unpack .../07-javascript-common_11_all.deb ...
Unpacking javascript-common (11) ...
Selecting previously unselected package libcupsfilters1:amd64.
Preparing to unpack .../08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb ...
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack .../09-libcupsimage2_2.2.7-1ubuntu2.8_amd64.deb ...
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.8) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../10-libijs-0.35_0.35-13_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-13) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../11-libjbig2dec0_0.13-6_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.13-6) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.14_all.deb
...
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.14_amd64.deb ...
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libjs-jquery.
Preparing to unpack .../14-libjs-jquery_3.2.1-1_all.deb ...
Unpacking libjs-jquery (3.2.1-1) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libpotrace0.
Preparing to unpack .../16-libpotrace0_1.14-2_amd64.deb ...
Unpacking libpotrace0 (1.14-2) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../18-rubygems-integration_1.11_all.deb ...
Unpacking rubygems-integration (1.11) ...
Selecting previously unselected package ruby2.5.
Preparing to unpack .../19-ruby2.5_2.5.1-1ubuntu1.10_amd64.deb ...
Unpacking ruby2.5 (2.5.1-1ubuntu1.10) ...
Selecting previously unselected package ruby.
Preparing to unpack .../20-ruby_1%3a2.5.1_amd64.deb ...


```

```

Unpacking ruby (1:2.5.1) ...
Selecting previously unselected package rake.
Preparing to unpack .../21-rake_12.3.1-1ubuntu0.1_all.deb ...
Unpacking rake (12.3.1-1ubuntu0.1) ...
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack .../22-ruby-did-you-mean_1.2.0-2_all.deb ...
Unpacking ruby-did-you-mean (1.2.0-2) ...
Selecting previously unselected package ruby-minitest.
Preparing to unpack .../23-ruby-minitest_5.10.3-1_all.deb ...
Unpacking ruby-minitest (5.10.3-1) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../24-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-power-assert.
Preparing to unpack .../25-ruby-power-assert_0.3.0-1_all.deb ...
Unpacking ruby-power-assert (0.3.0-1) ...
Selecting previously unselected package ruby-test-unit.
Preparing to unpack .../26-ruby-test-unit_3.2.5-1_all.deb ...
Unpacking ruby-test-unit (3.2.5-1) ...
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack .../27-libruby2.5_2.5.1-1ubuntu1.10_amd64.deb ...
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.10) ...
Selecting previously unselected package libsyntax1:amd64.
Preparing to unpack .../28-libsyntax1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libsyntax1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack .../29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
.../30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libzip-0-13:amd64.
Preparing to unpack .../31-libzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb ...
Unpacking libzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../32-lmodern_2.004.5-3_all.deb ...
Unpacking lmodern (2.004.5-3) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../33-preview-latex-style_11.91-1ubuntu1_all.deb ...
Unpacking preview-latex-style (11.91-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../34-t1utils_1.41-2_amd64.deb ...
Unpacking t1utils (1.41-2) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../35-tex-gyre_20160520-1_all.deb ...

```

```
Unpacking tex-gyre (20160520-1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../37-texlive-base_2017.20180305-1_all.deb ...
Unpacking texlive-base (2017.20180305-1) ...
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../38-texlive-fonts-recommended_2017.20180305-1_all.deb ...
Unpacking texlive-fonts-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../39-texlive-plain-generic_2017.20180305-2_all.deb ...
Unpacking texlive-plain-generic (2017.20180305-2) ...
```

```
[ ]:  jupyter nbconvert --to pdf --output "3rd_sem1" MTP_BNN.ipynb
```

```
[ ]:
```

```
[ ]:
```

```
[ ]:
```