**(Course Name) Project**

Name Surname - Student ID

## Problem Tanımı

Belirli bir makinenin arızalanıp arızalanmayacağının yanı sıra arızanın niteliğini belirleme ihtiyacı, 4.0 kuşağı endüstrileri için çok önemlidir. Bunun ana nedeni şu düşüncenin altında yatmaktadır: Arızalı bir makinenin onarımı veya değiştirilmesi, genellikle tek bir bileşenin değiştirilmesi için gerekenden çok daha yüksek maliyetler gerektirir. Bu nedenle, makinelerin durumunu izleyen sensörlerden gelen verilerin analiz edilmesi büyük tasarruf sağlayabilir.

## Amaç

Bu kapsamda University of California Center for Machine Learning and Intelligent Systems tarafından yayınlanmış ve endüstride karşılaşılan gerçek öngörücü bakım uygulamalarını yansıtan sentetik olarak oluşturulmuş veri seti üzerinde keşifti yöntemlerle analizler yapılacaktır.

Orijinal veri kaynağı:

<https://archive.ics.uci.edu/ml/datasets/AI4I+2020+Predictive+Maintenance+Dataset>

İndirilen kaynak:

<https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification?datasetId=1697740&sortBy=voteCount>

Veri seti hakkında:

Veri seti 14 özelliğe sahip 10.000 veri noktasından oluşur. Bu özellikler:

* UDI: (unique device identifier) 1 ile 10000 arasında değişen sıralayıcı kod
* Product ID: ürüne özel benzersiz tanımlayıcı kod
* Type: farklı ürün kalitelerini temsil eden L, M veya H harflerinden oluşan kategorik değer
* Air Temperature [K]: 300 K bandında civarında 2 K standart sapmayla normalleştirilerek üretilmiş rastgele sıcaklık değerleri
* Process Temperature [K]: hava sıcaklığına artı 10 K eklenerek 1 K standart sapmayla normalleştirilerek üretilmiş rastgele sıcaklık değerleri
* Rotational speed [rpm]: 2860 W’lık güç değeri üzerinden hesaplanmış dönüş hızları
* Torque [nm]: 40 nm aralığında ve f (frictional torque) = 10 Nm ile negatif olmayan değerlerle dağıtılmıştır
* Tool wear [min]: H/M/L kalite varyantları, süreçte kullanılan takıma 5/3/2 dakikalık aşınma ekler
* Failure type: Makine arızası, beş bağımsız arıza türünden oluşur: Tool wear failure (TWF), heat dissipation failure (HDF) , power failure (PWF), overstrain failure (OSF) ve herhangi bir değişkenle alakasız olarak oluşmuş random failure türü.
* Target, herhangi bir arıza modu için makinenin bu belirli veri noktasında başarısız olup olmadığını gösterir. Yukarıdaki arıza türlerinden en az biri doğruysa, işlem başarısız olur ve "Target" 1 olarak ayarlanır.

## Keşifçi Veri Analizi

##gerekli kütüphanelerin import edilmesi

import pandas as pd

import numpy as np

import seaborn as sns

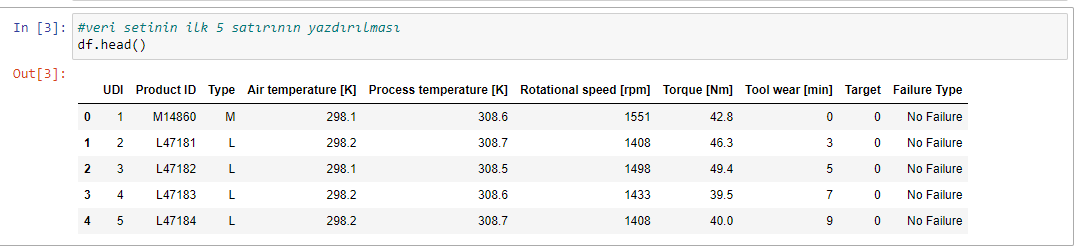
import matplotlib.pyplot as plt

#veri setinin yüklenmesi

df = pd.read\_csv("predictive\_maintenance.csv")

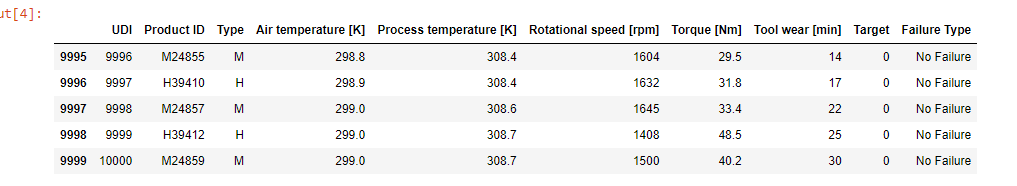
#veri setinin ilk 5 satırının yazdırılması

df.head()

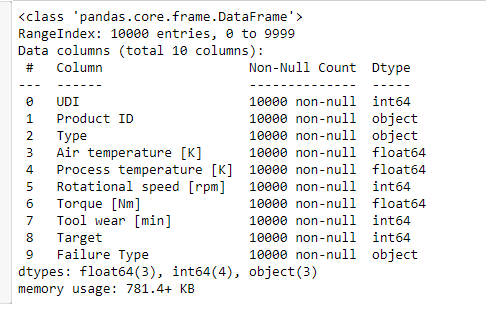


#veri setinin son 5 satırının yazdırılması

df.tail()

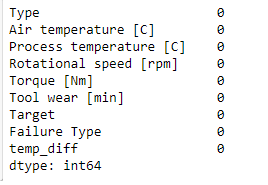


df.info()



df.isnull().sum()

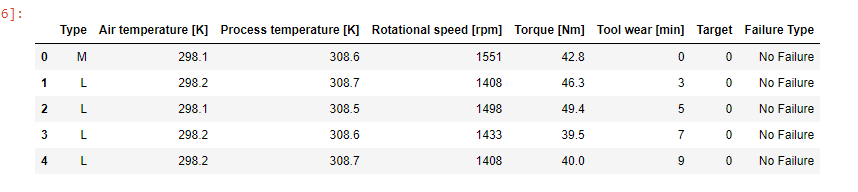
#veri setinde boş alan yoktur



#kullanılmayacak UDI ve Product ID Satırlarının silinmesi

df.drop(['UDI','Product ID'], axis=1, inplace=True)

df.head()



#Kelvin olarak verilen sıcaklığın celcius santigrat dereceye çevrimi

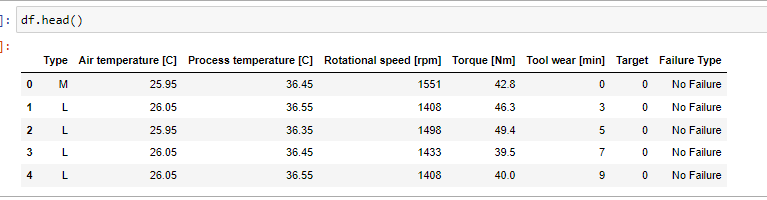
df["Air temperature [K]"] = df["Air temperature [K]"] - 272.15

df["Process temperature [K]"] = df["Process temperature [K]"] - 272.15

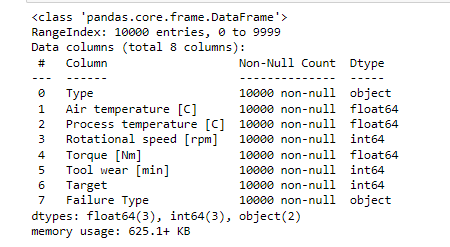
# Sütun adlarındaki Kelvinlerin (K) Centigrate(C) olarak değiştirilmesi

df.rename(columns={"Air temperature [K]" : "Air temperature [C]","Process temperature [K]" : "Process temperature [C]"},inplace=True)

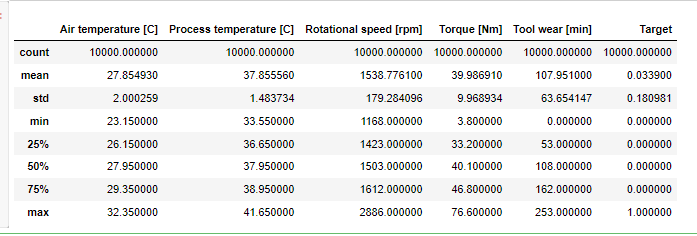
df.head()



df.info()

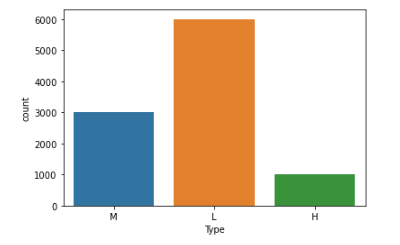


df.describe()



#ürün türüne göre dağılım

ax = sns.countplot(x="Type", data=df)

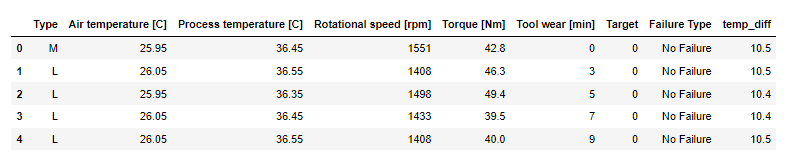


#sıcaklık farkı isimli yeni bir sütun oluşturulması ve ilk 5 satırın yazdırılması

# temp\_diff = Process temperature - Process temperature

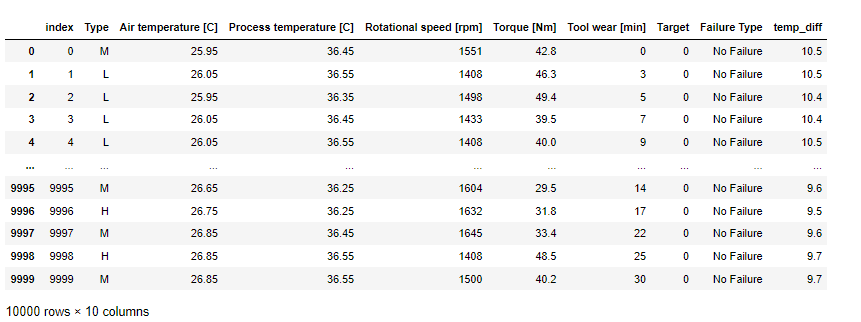
df['temp\_diff'] = pd.DataFrame(df['Process temperature [C]']-df['Air temperature [C]'])

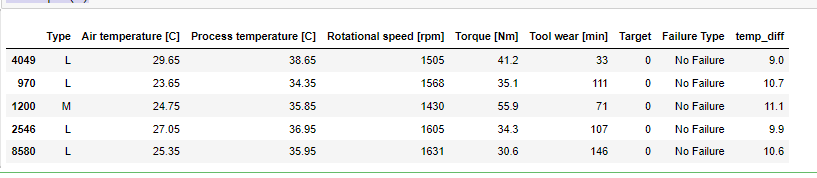
df.head()



##indeksin resetlenerek sütun haline getirilmesi

df.reset\_index()



df.sample(5)

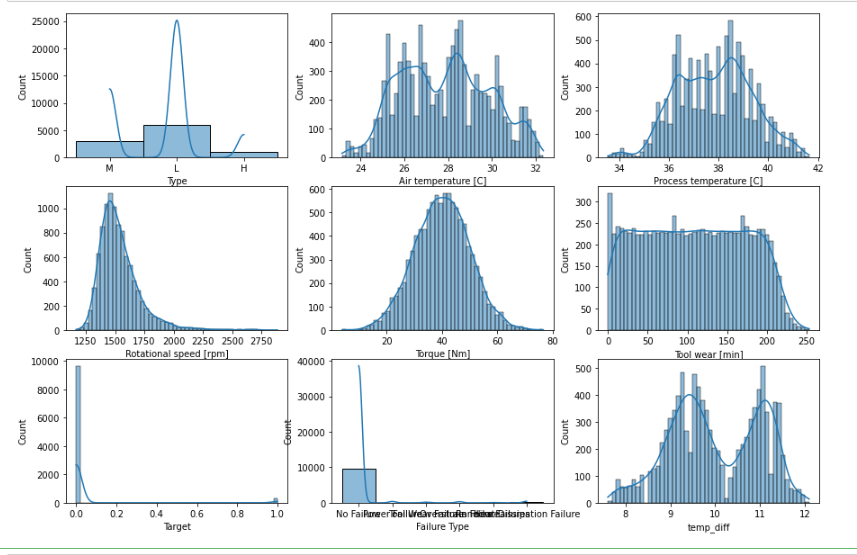
#veri setindeki tüm değişkenlerin histogram dağılımının yapılması

plt.figure(figsize=(15,10))

for i,col in enumerate(df.columns,1):

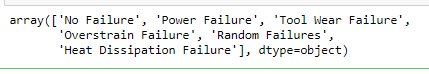
plt.subplot(3,3,i)

sns.histplot(df[col],kde=True, bins=50)



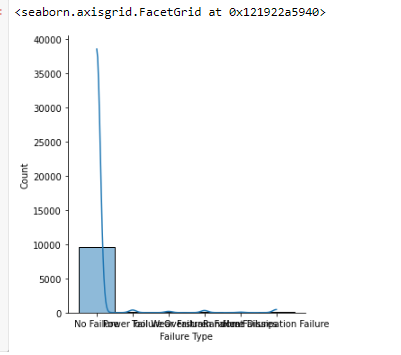
#Hangi tür arıza tipleri olduğunu görmek için

df['Failure Type'].unique()



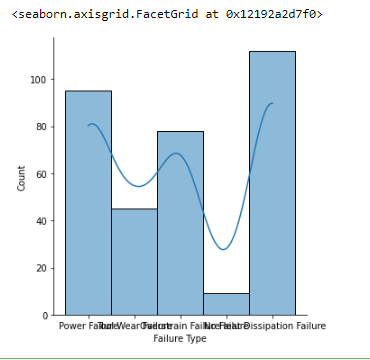
#seçilen tek değişkenin dağılımının basılması

sns.displot(data=df, x="Failure Type", kde=True)



#seçilen tek değişkenin dağılımının basılması (sadece arıza olduğu durumda yani Target=1'ken)

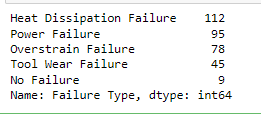
sns.displot(data=df[df['Target'] == 1], x="Failure Type", kde=True)



#4 tip arıza beklerken 5 tip basılmış, dolayısıyle Arıza türlerinin incelenmesi gerekmektedir

df1=df[df['Target'] == 1]

df1['Failure Type'].value\_counts()



#Target =1 ken yani arıza var ise arıza tipinde no failure olmamalı. Bu veriler yanlış işaretlenmiş

#yanlış işaretlenen verilerin silinmesi

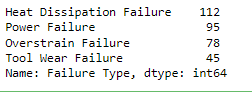
indexNames = df[(df['Target'] == 1) & (df['Failure Type'] == 'No Failure')].index

df.drop(indexNames , inplace=True)

##yanlış işaretlenen verilerin silinmesinin kontrolü

df2=df[df['Target'] == 1]

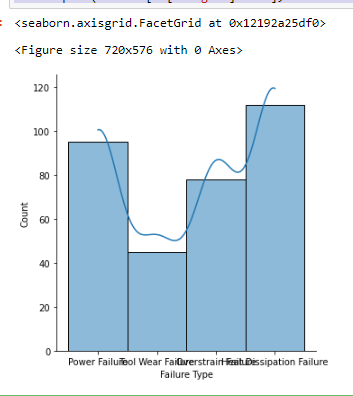
df2['Failure Type'].value\_counts()



#seçilen tek değişkenin dağılımının basılması

plt.figure(figsize = (10, 8))

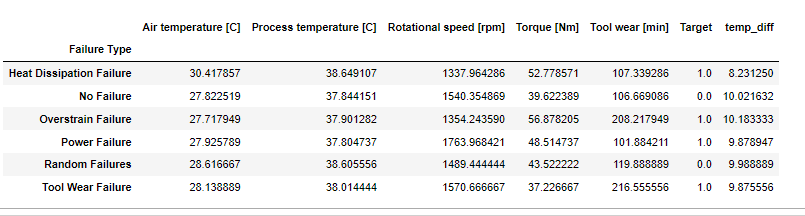
sns.displot(data=df[df['Target'] == 1], x="Failure Type", kde=True)



##veri setinin arıza türü bazında gruplanması ve değerlerin ortalaması

arizaturu = df.groupby('Failure Type').mean()

arizaturu



labels\_Failed = ["M", "L", "H"]

#türlere göre arızaların ayrılması

M\_Failed = sum(df.loc[df['Type']=='M'].Target)

L\_Failed = sum(df.loc[df['Type']=='L'].Target)

H\_Failed = sum(df.loc[df['Type']=='H'].Target)

Failed=[M\_Failed, L\_Failed, H\_Failed]

#Kalite türlerine göre toplam ürün sayıları

M\_Tot = len(df.loc[df['Type']=='M'].Target)

L\_Tot = len(df.loc[df['Type']=='L'].Target)

H\_Tot = len(df.loc[df['Type']=='H'].Target)

#türlere göre sorunsuz olanların ayrılması

M\_NF = M\_Tot-M\_Failed

L\_NF = L\_Tot-L\_Failed

H\_NF = H\_Tot-H\_Failed

NFail = [M\_NF, L\_NF, H\_NF]

fig, ax = plt.subplots(1,1)

width = 0.3

ax.bar(labels\_Failed, Failed, width, label='Arızalı',color='Red')

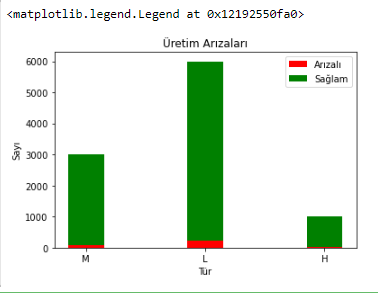
ax.bar(labels\_Failed, NFail, width, bottom=Failed,label='Sağlam',color='green')

ax.set\_xlabel('Tür')

ax.set\_ylabel('Sayı')

ax.set\_title('Üretim Arızaları')

ax.legend()



#hata yüzdelerinin basılması

print('Sağlam ürünler:',round((M\_NF+L\_NF+H\_NF)\*100/(M\_Tot+L\_Tot+H\_Tot),1),'%')

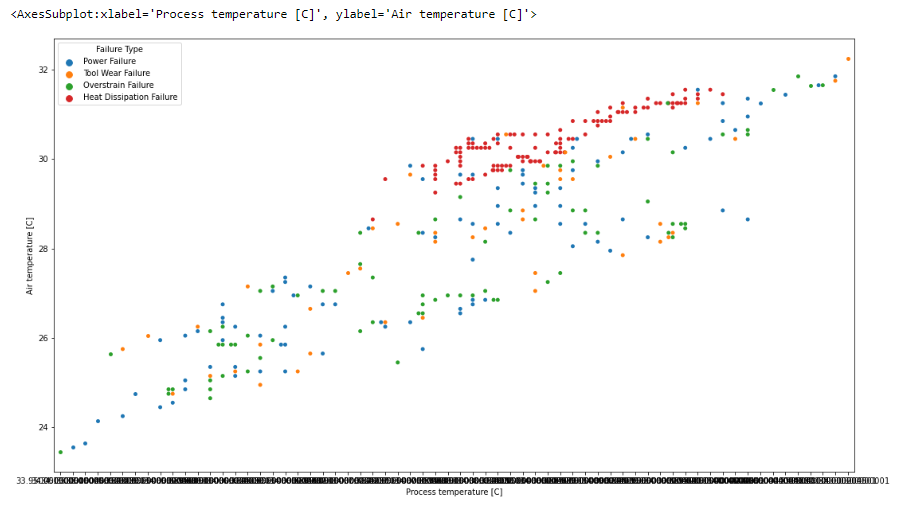
print('Arızalı ürünler:',round((M\_Failed+L\_Failed+H\_Failed)\*100/(M\_Tot+L\_Tot+H\_Tot),1),'%')



#Sıcaklık eksenlerine göre sürü grafiği

plt.figure(figsize=(18,10))

sns.swarmplot(data=df[df['Target'] == 1],x="Process temperature [C]",y='Air temperature [C]',hue="Failure Type")



#Sürü grafiği farklı kategorik değişkenlerin görselleştirilmesine yardımcı olur.

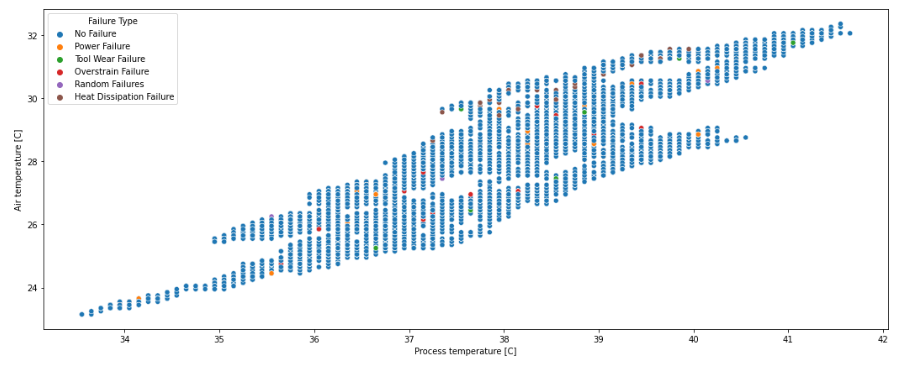
#benzer şekilde scatter plot (serpilme grafiği) de 2 tür verinin incelendiği durumda kullanılabilir,

#bu veri analizi için sürü grafiğine göre daha hızlı sonuç alınmıştır.

#farklı değişkenlere göre arızaların oluşmasının gözlemlenmesi (arıza olmayan durumlar hariç tutulmadan)

plt.figure(figsize=(18,7))

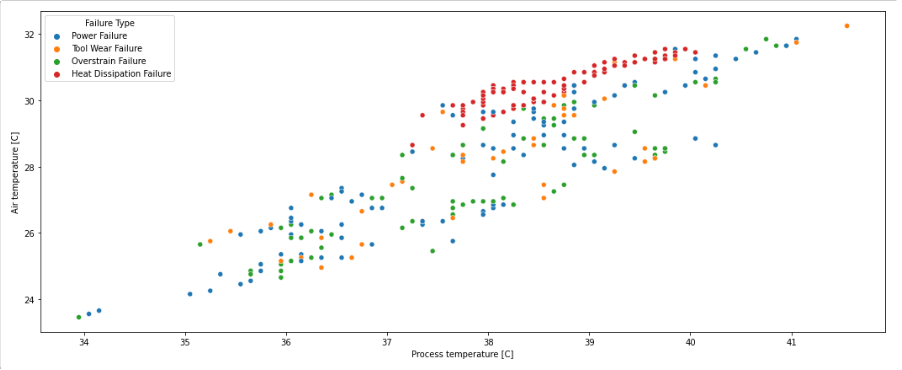
sns.scatterplot(data=df, x="Process temperature [C]", y="Air temperature [C]", hue="Failure Type");



#farklı değişkenlere göre arızaların oluşmasının gözlemlenmesi (arızalı olduğu durumda yani df['Target'] == 1 iken)

plt.figure(figsize=(18,7))

sns.scatterplot(data=df[df['Target'] == 1], x="Process temperature [C]", y="Air temperature [C]", hue="Failure Type");



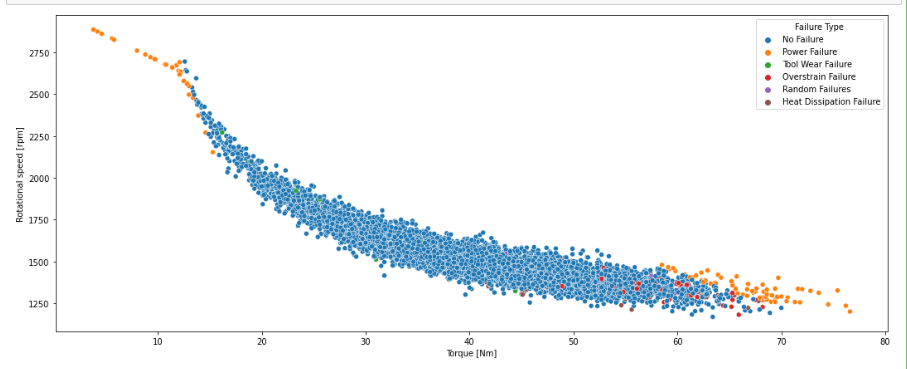
#bazı durumlarda verilerin belirli bir kısmının analiz edilmesi daha anlaşılır sonuçlar sunabilirken

#bazılarında tüm verinin değerlendirilmesi daha açıklayıcı olabilir.

#farklı değişkenlere göre arızaların oluşmasının gözlemlenmesi (arıza olmayan durumlar hariç tutulmadan) - Torque ve rotational speed için

plt.figure(figsize=(18,7))

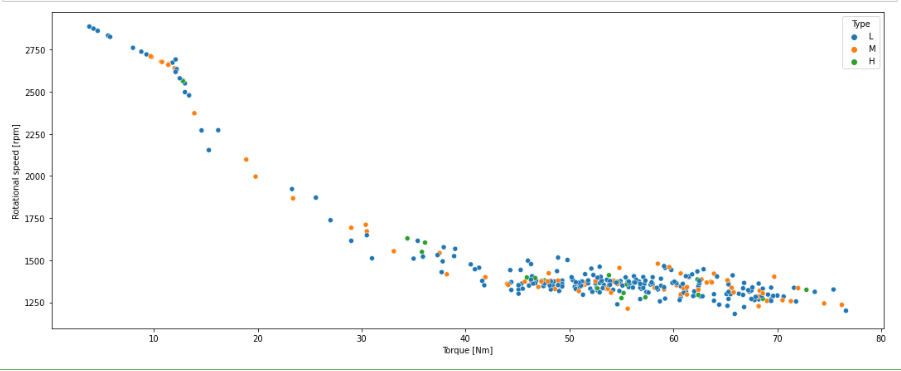
sns.scatterplot(data=df, x="Torque [Nm]", y="Rotational speed [rpm]", hue="Failure Type");



# Torque ve rotational speed değerlerine göre arıza olan durumda (Target = 1 iken) ürün türü dağılımı

plt.figure(figsize=(18,7))

sns.scatterplot(data=df[df['Target'] == 1], x="Torque [Nm]", y="Rotational speed [rpm]", hue="Type");



## Sonuç

* Heat dissipation failure (ısı dağılımı arızası) düşük RPM'de ve daha yüksek torkta meydana gelir. Sıcaklık grafiğine bakıldığında hava ve süreç sıcaklık farkının düşük olduğu aralıklarda gerçekleştiği de görülmektedir. Bu iki bilgiyi birleştirdiğimizde Heat dissipation arızasının makinenin ilk çalıştığı zaman aralığında daha sık meydana geldiğini söyleyebiliriz.
* Overstrain failure (Aşırı gerilme arızası), daha yüksek torkta ve daha düşük RPM'de gerçekleşir, ancak ısı dağılımı arızasının aksine, sıcaklık farkından bağımsızdırlar.
* Power failures (Güç kesintileri) ya yüksek tork ve düşük devir durumunda ya da yüksek devir ve düşük tork durumunda meydana gelir. Bu iki değişkenin dengesinin korunması arızayı önlemek için önemli bir etken kabul edilebilir.
* Random failure durumları arıza olmayan durumda yani Target = 0 ‘ken gerçekleşmiştir. Veri seti tanımında bu arızaların herhangi bir sebepten olabileceği yazılmıştır. Ancak Target=0 durumunda görülmeleri hatalı işaretleme kabul edilebilir.
* Tool wear değişkeni veri seti tanımında sadece “H/M/L kalite varyantları, süreçte kullanılan araca 5/3/2 dakikalık takım aşınması ekler.” şeklinde tanımlanmıştır. Bu tanımın veri setine yansıması anlaşılamadığından Tool wear failure (TWF) hataları değerlendirilememiştir.
* Sadece keşifçi veri analizi yöntemleri kullanılarak bile önleyici bakım yaklaşımında kullanılabilecek öngörüler elde edilebilmektedir.