Machine Learning-Powered Equipment Failure Prediction System

Beginning

Overview

Business Understanding

Manufacturing facilities, telecommunications networks, and essential service providers face a universal operational challenge: critical equipment failures that disrupt core business functions and generate substantial financial and operational consequences. Manufacturing plants struggle with production machine breakdowns that halt assembly lines and delay order fulfillment. Telecommunications companies face antenna and tower equipment failures that disrupt network coverage and service reliability. Service businesses including banks, schools, and hospitals confront power generator failures during outages that threaten their ability to operate and serve their communities. These equipment failures create cascading effects that impact revenue, safety, and service delivery across multiple sectors. Each stakeholder group shares the fundamental challenge of maintaining operational continuity despite aging equipment and unpredictable failure patterns that threaten their primary business functions.

While advanced sensor systems provide real-time monitoring, many organizations can begin their predictive maintenance journey through structured manual data collection. Maintenance teams can document repair histories, component replacements, and performance degradation observations. This manually collected data, when systematically gathered over time, provides the foundation for predictive insights without requiring immediate capital investment in IoT infrastructure. For this demonstration project, we utilize a comprehensive dataset that mirrors the type of information organizations can collect through these practical methods.

The predictive model we've developed demonstrates how organizations can leverage their existing data collection processes to achieve significant operational improvements. By analyzing historical equipment performance data similar to what companies already track, our solution identifies failure patterns and provides actionable early warnings. This approach proves particularly valuable for businesses with critical but not extensively instrumented equipment, showing how systematic data collection and analysis can transform maintenance from reactive to predictive, delivering tangible financial benefits without massive upfront technology investments.

The predictive maintenance approach delivers substantial cost minimization through intelligent failure prediction, regardless of data collection methodology. True positives enable organizations to schedule maintenance during planned downtime, avoiding expensive emergency repairs and operational disruptions. False positives represent the minimal cost of precautionary maintenance,

which remains significantly lower than unexpected downtime expenses. False negatives carry the highest cost through unplanned operational halts and emergency response requirements. By optimizing this balance, businesses achieve meaningful cost reduction while building the business case for more advanced monitoring systems, creating a virtuous cycle of continuous improvement in equipment reliability and operational efficiency.

Industry statistics overwhelmingly support the business case for predictive maintenance. According to Deloitte research, unplanned downtime costs industrial manufacturers approximately 50 billion dollars annually, while companies implementing predictive maintenance achieve 70-75 percent reductions in equipment failures (McKinsey) and 25-30 percent decreases in maintenance costs. The financial impact is substantial, with manufacturing plants losing 22,000 dollars per minute during production stoppages (Automotive Manufacturing Association) and telecommunications outages costing 15,000-20,000 dollars per minute (ITIC Research). Predictive approaches deliver remarkable ROI, with average returns of 10x (Capgemini Research Institute) and payback periods of 6-9 months (Bain & Company), while optimizing maintenance strategies can reduce costs by 45-55 percent compared to reactive approaches and increase equipment availability by 9 percent (Plant Engineering metrics).

Operational resilience is no longer an aspiration but a financial imperative. With the cost of downtime measured in tens of thousands per minute, the transition from reactive maintenance to a predictive, intelligence-driven strategy is the single most impactful lever an organization can pull to safeguard profitability, ensure business continuity, and secure a decisive competitive advantage in today's market.

Data Understanding

Data Source and Properties

The dataset utilized for this predictive maintenance project is a comprehensive synthetic dataset containing **10,000 historical records** of machine operations with **10 relevant features** that directly mirror real-world industrial equipment sensor data. This dataset provides a robust foundation for developing predictive models applicable to manufacturing machinery, telecommunications infrastructure, and power generation systems.

Dataset Composition and Feature Relevance

Core Features with Direct Business Relevance:

- **`Type (L/M/H)`**: Represents product quality variants, simulating different equipment models or configurations found in real fleets
- **`Air temperature [K] & Process temperature [K]`** : Critical thermal monitoring parameters applicable to all industrial equipment
- **`Rotational speed [rpm]`** : Fundamental for motors, engines, and rotating machinery across industries

- `Torque [Nm]` : Direct mechanical load measurement essential for detecting overloading conditions
- `Tool wear [min]` : Cumulative degradation metric representing equipment aging and wear patterns

Descriptive Statistics Demonstrating Real-World Utility

Statistical Overview:

- **Dataset Size**: 10,000 records providing substantial statistical power for model development
- **Failure Distribution**:
  - **Overall Failure Rate**: 3.4% (339 failure events) - realistic for industrial equipment
  - **Failure Types**: Heat Dissipation (112), Power (95), Overstrain (78), Tool Wear (45), Random (18)
- **Operational Ranges**:
  - Temperature: 295.3K - 304.5K (covering normal to stress conditions)
  - Rotational Speed: 1,168 - 2,886 rpm (idle to maximum operational range)
  - Torque: 3.8 - 76.6 Nm (light load to overload scenarios)
  - Tool Wear: 0 - 253 minutes (new to heavily worn equipment)

Feature Justification for Predictive Maintenance

Type Classification: The L/M/H categorization (60%/30%/10% distribution) enables model development across equipment tiers, simulating real-world scenarios where organizations maintain mixed equipment fleets with varying reliability characteristics.

Temperature Features: The 9K operational range (295.3K-304.5K) captures realistic thermal variations, enabling detection of overheating conditions critical for preventing electrical failures and thermal stress damage.

Rotational Speed: The broad range (1,168-2,886 rpm) spans idle to maximum operational speeds, allowing identification of abnormal speed patterns that precede mechanical failures in motors and rotating components.

Torque Measurements: The wide torque spectrum (3.8-76.6 Nm) facilitates detection of both under-loading and overloading conditions, crucial for identifying transmission problems and mechanical stress.

Tool Wear: The progressive wear metric (0-253 minutes) provides cumulative damage assessment, enabling proactive replacement scheduling and lifespan optimization.

Data Limitations and Implications

Key Limitations:

1. **Class Imbalance**: The 3.4% failure rate creates a significant class imbalance challenge requiring specialized sampling techniques
2. **Failure Distribution**: Some failure types have limited examples (only 18 Random Failures) affecting model performance on rare events
3. **Synthetic Nature**: While comprehensive, synthetic data may not capture all real-world operational complexities and noise patterns
4. **Temporal Gaps**: Lacks continuous time-series context which would be available in real monitoring systems

Implications for Project Scope: These limitations define the current scope as a **proof-of-concept demonstration** that validates the analytical approach. The dataset successfully captures essential failure patterns and operational relationships, providing a transferable foundation that can be extended with organization-specific data during implementation phases.

The comprehensive feature set and realistic failure distributions demonstrate strong suitability for developing predictive maintenance models that address real-world business challenges across multiple industrial sectors.

Data Preparation

Importing libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import plotly.express as px
from sklearn.preprocessing import OrdinalEncoder, MinMaxScaler, RobustScaler
from sklearn.model_selection import GridSearchCV, RandomizedSearchCV, cross_val_score,
from sklearn.ensemble import RandomForestClassifier, BaggingClassifier
from imblearn.ensemble import BalancedRandomForestClassifier, BalancedBaggingClassifier
from imblearn.over_sampling import SMOTE, ADASYN, RandomOverSampler, BorderlineSMOTE, S
from imblearn.under_sampling import TomekLinks, RandomUnderSampler, NearMiss, ClusterCe
from sklearn.pipeline import Pipeline
from sklearn.metrics import accuracy_score, classification_report, precision_score, f1_
from sklearn.metrics import precision_recall_fscore_support, roc_curve, precision_recal
import pickle
sns.set()
```

In [3]:
```
raw_data = pd.read_csv('predictive_maintenance.csv')
df = raw_data.copy()
df.head()
```

Out[3]:

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	No Failure

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Failure Type
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	No Failure
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	No Failure
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	No Failure
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	No Failure

In [4]:
```
df.describe()
```

Out[4]:

	UDI	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target
count	10000.00000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	5000.50000	300.004930	310.005560	1538.776100	39.986910	107.951000	0.033900
std	2886.89568	2.000259	1.483734	179.284096	9.968934	63.654147	0.180981
min	1.00000	295.300000	305.700000	1168.000000	3.800000	0.000000	0.000000
25%	2500.75000	298.300000	308.800000	1423.000000	33.200000	53.000000	0.000000
50%	5000.50000	300.100000	310.100000	1503.000000	40.100000	108.000000	0.000000
75%	7500.25000	301.500000	311.100000	1612.000000	46.800000	162.000000	0.000000
max	10000.00000	304.500000	313.800000	2886.000000	76.600000	253.000000	1.000000

Quick insights:

There are 10.000 instances in the dataset

The are no missing values (every feature has 10.000 observations)

Target variable are ones and zeros

UDI seems to be an index number

Product ID is an identification number.

In [5]:
```
# Dropping 'UDI' and 'Product ID' from the dataset
df.drop(['UDI', 'Product ID'], axis=1, inplace=True)
```

In [6]:
```
# Taking a look at 'Failure Type' and 'Target' variable
df['Failure Type'].value_counts()
```

Out[6]:
```
No Failure                 9652
Heat Dissipation Failure    112
```

```
Power Failure                95
Overstrain Failure           78
Tool Wear Failure            45
Random Failures              18
Name: Failure Type, dtype: int64
```

This target variable assumes six possible values: no failure, or five different types of failure.

We can see that the dataset is highly unbalanced.

In [7]:
```
df['Target'].value_counts(normalize=True)
```

Out[7]:
```
0    0.9661
1    0.0339
Name: Target, dtype: float64
```

Even there, the dataset is unbalanced.

In [8]:
```
# Rechecking for missing values
df.isna().sum()
```

Out[8]:
```
Type                       0
Air temperature [K]        0
Process temperature [K]    0
Rotational speed [rpm]     0
Torque [Nm]                0
Tool wear [min]            0
Target                     0
Failure Type               0
dtype: int64
```

In [9]:
```
# Taking a look at the data types
df.dtypes
```

Out[9]:
```
Type                        object
Air temperature [K]        float64
Process temperature [K]    float64
Rotational speed [rpm]       int64
Torque [Nm]                float64
Tool wear [min]              int64
Target                       int64
Failure Type                object
dtype: object
```

'Type' and 'Failure Type' are string variable.

Target variables

There are two target variables: 'Target' and 'Failure Type'. we are going to check if there is no inconsistancies.

In [10]:
```
# Checking types of failure
df['Failure Type'].value_counts()
```

Out[10]:
```
No Failure                 9652
Heat Dissipation Failure    112
Power Failure                95
Overstrain Failure           78
Tool Wear Failure            45
Random Failures              18
Name: Failure Type, dtype: int64
```

In [11]:
```
df_failure = df[df['Target'] == 1]
df_failure['Failure Type'].value_counts()
```

Out[11]:
```
Heat Dissipation Failure    112
Power Failure                95
Overstrain Failure           78
Tool Wear Failure            45
No Failure                    9
Name: Failure Type, dtype: int64
```

9 values that are classified a failure in the 'Target' variable are classified as no failure in the 'Failure Type' variable. We can't tell wether they are failure or no failure so we will remove them.

In [12]:
```
index_possible_failure = df_failure[df_failure['Failure Type'] == 'No Failure'].index
df.drop(index_possible_failure, axis=0, inplace=True)
```

So now we will do the same with the target variable equal to 0, no failure. We will see how many failure types are misplaced.

In [13]:
```
df_failure = df[df['Target'] == 0]
df_failure['Failure Type'].value_counts()
```

Out[13]:
```
No Failure         9643
Random Failures      18
Name: Failure Type, dtype: int64
```

So there is 18 misplaced values. We will remove them.

In [14]:
```
random_failure_indices = df[df['Failure Type'] == 'Random Failures'].index
df.drop(random_failure_indices, axis=0, inplace=True)
```

27 instances were removed (0.27% of the entire dataset). Of which:

9 belonged to class Failure in 'Target' variable and No failure in target 'Failure Type' 18 belonged to class No failure in 'Target' variable and Random failures in target 'Failure Type'

In [15]:
```
# We can check that all 27 instances were removed from the dataset:
df.shape[0]
```

Out[15]:
```
9973
```

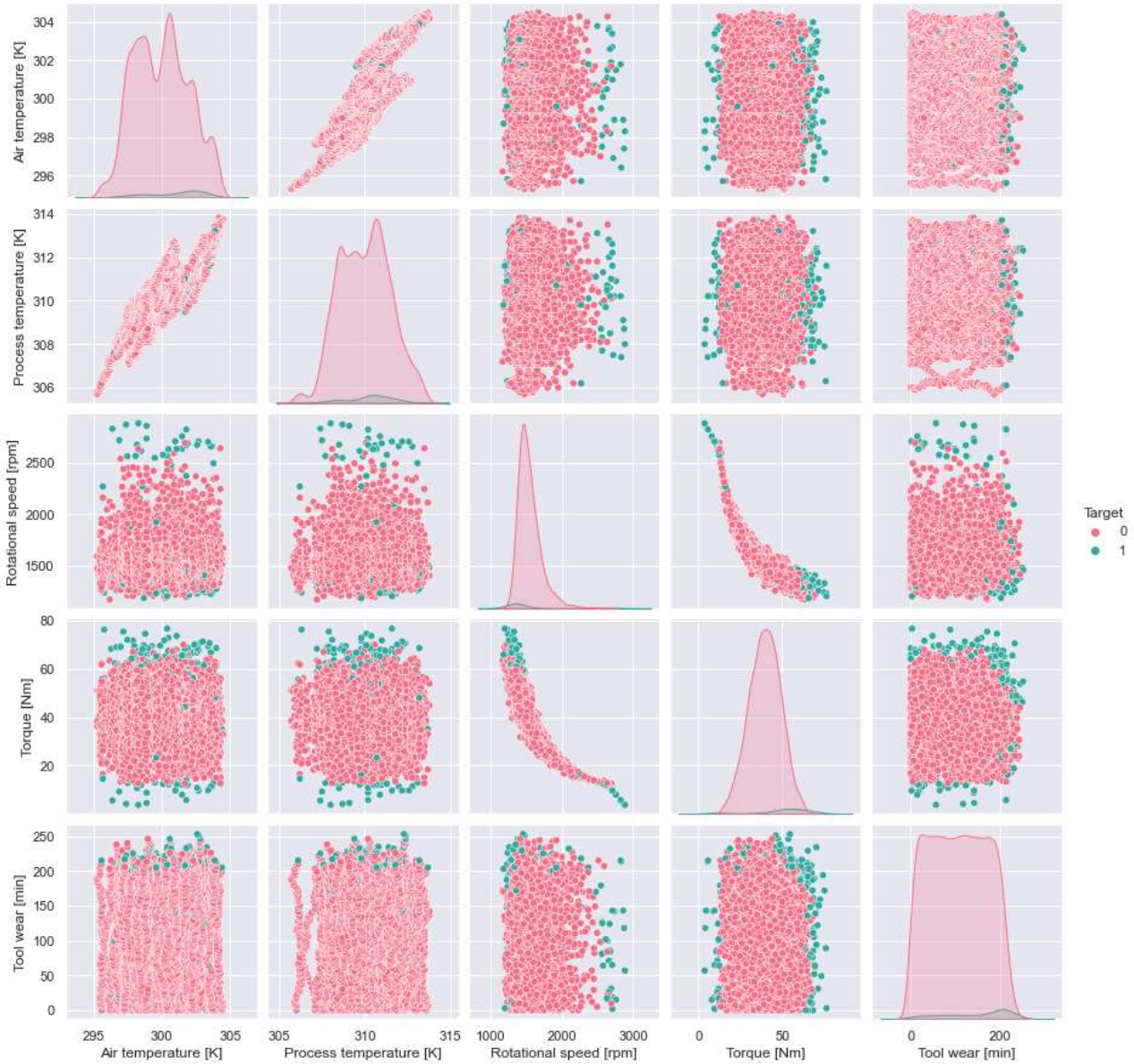
In [16]:
```
# reset the index
df.reset_index(inplace=True, drop=True)
```

EDA

Correlation between the main variables

In [17]:
```
sns.pairplot(df, hue='Target', palette='husl')
```

Out[17]:
```
<seaborn.axisgrid.PairGrid at 0x1fc9d9a3c70>
```



Observations: In the context of the predictive maintenance dataset, the strong negative correlation between Torque and Rotational Speed is a fundamental principle in mechanics: when a machine is performing a demanding task (high torque), its speed naturally drops. Conversely, when it is running fast (high rpm), it is facing low resistance (low torque). This inverse relationship is a sign of a correctly functioning, power-limited system.

The strong positive correlation between Process Temperature and Air Temperature indicates that the machine's internal operating temperature is heavily influenced by the ambient environment.

We immediately see that failures occur for extreme values of some features, i.e., the machinery fails either for the lowest or largest values of torque and rotational speed. This is easily spotted in the graph since the green dots are far apart for those features. So, there is a range for normal conditions in which the machines operate, and above or under this range, they tend to fail.

Percentage of failure

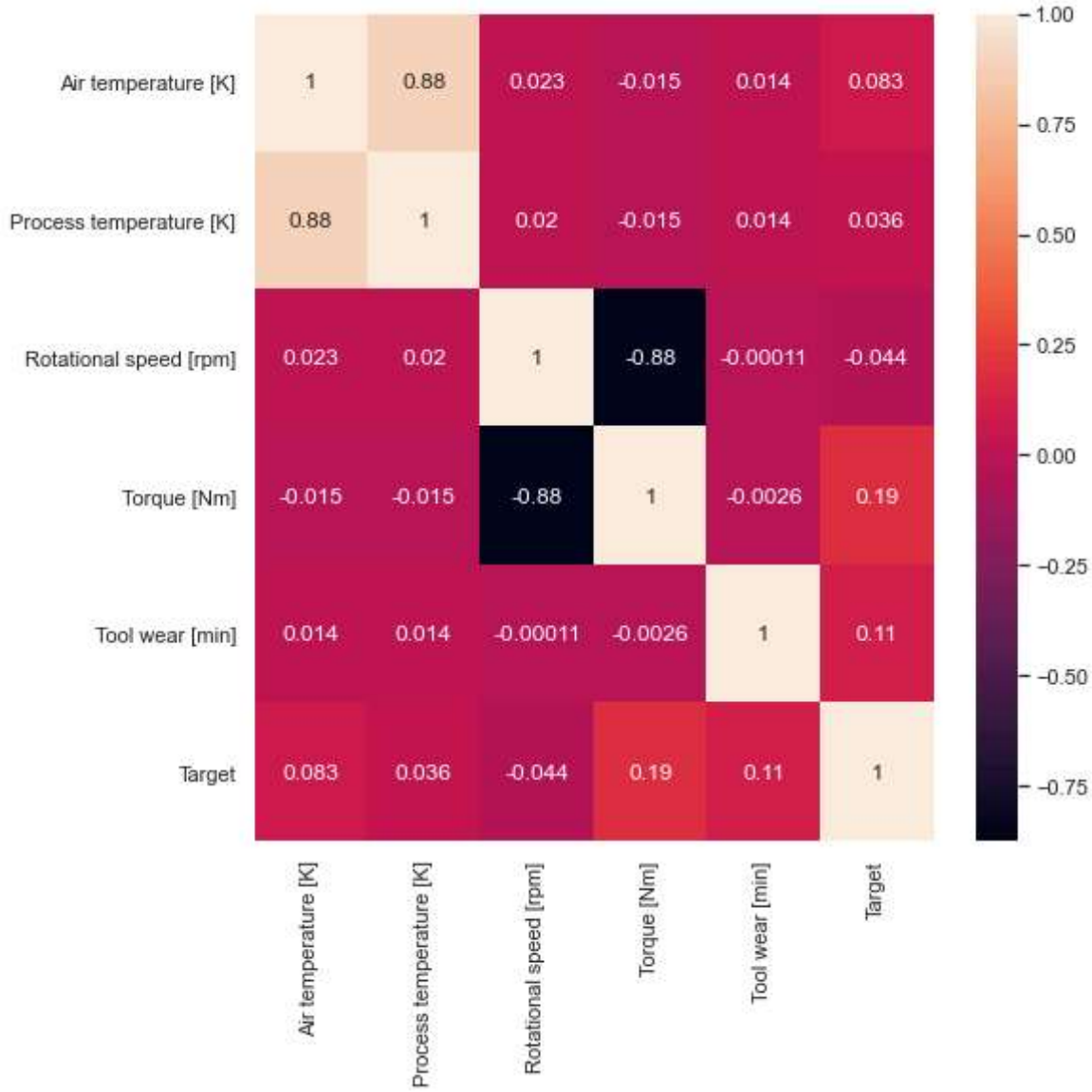
In [18]:
```
colors = ['#E1728F', '#409E7D']
plt.pie(df['Target'].value_counts(),  explode=[0.1, 0.2], labels=['Not failure', 'Failu
```

```
        autopct='%1.1f%%', wedgeprops={'edgecolor': 'black'}, shadow=True, startangle=2
        colors=colors)
plt.title('Failure vs not failure')
plt.tight_layout()
plt.show()
```

The data as we said before, is highly unbalanced.

Correlation heatmap

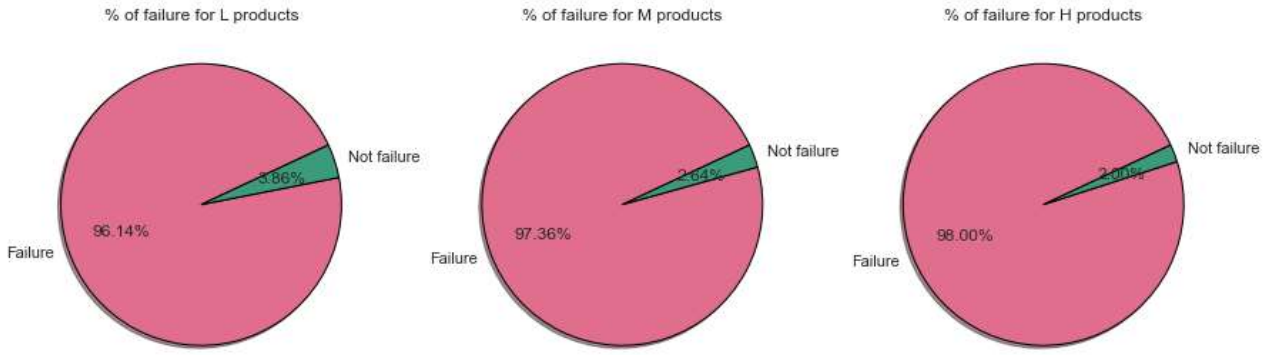
In [19]:
```
plt.figure(figsize=(8, 8))
sns.heatmap(df.corr(), annot=True)
plt.show()
```



As we said before, there is high negative correlation between rotational speed and Torque, and between process temperature and air temperature.

Percentage of failure per product type

In [20]:
```
fig, axes = plt.subplots(1,3, figsize=[15,5])
axes.flatten()
j=0
colors = ['#E1728F', '#409E7D']
for i in ['L', 'M', 'H']:
    df_product_type = df[df['Type'] == i]
    axes[j].pie(df_product_type['Target'].value_counts(), labels=['Failure', 'Not failu
                autopct='%1.2f%%', wedgeprops={'edgecolor': 'black'}, shadow=True, star
                colors=colors)
    axes[j].set_title('% of failure for ' + i + ' products')
    j+=1
```



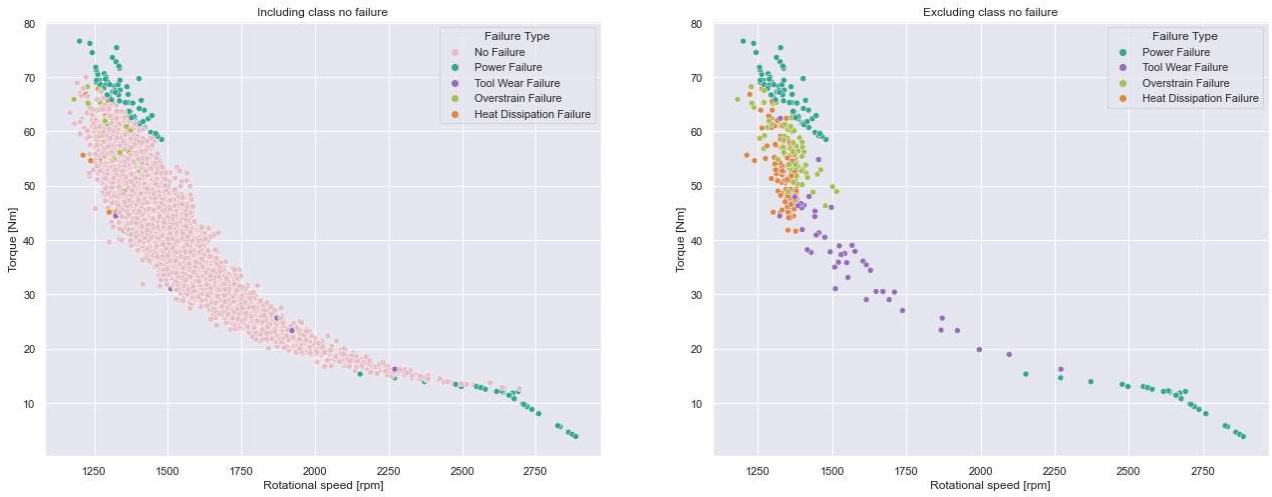
Observations: L products have a higher ratio of failure compared to the other product types. Moreover, M tends to fail more than H products, logically.

Exploring features for each type of failure

In [24]:
```
fig, ax = plt.subplots(1,2, figsize=[22,8])
plt.title('Rot. Speed vs Torque wrt Failure Type')
sns.scatterplot(data=df, x='Rotational speed [rpm]', y='Torque [Nm]', hue='Failure Type
sns.scatterplot(data=df[df['Target'] == 1], x='Rotational speed [rpm]', y='Torque [Nm]'

ax[0].set_title('Including class no failure')
ax[1].set_title('Excluding class no failure')
```

Out[24]:
```
Text(0.5, 1.0, 'Excluding class no failure')
```



The first plot, "Including class no failure," clearly illustrates the strong negative correlation we observed earlier: The vast majority of data points (No Failure, represented by the light pink/gray color) form a distinct, sweeping curve: as Rotational Speed increases, Torque decreases, and vice-versa.

The second plot, "Excluding class no failure," removes the dominating No Failure points, making the failure events stand out clearly and revealing the conditions that lead to different types of machine failure.

A. Power Failure (Purple dots)Condition: These failures occur at very low Rotational Speeds (mostly below $1400\ \mathrm{rpm}$) and a wide range of high Torques (mostly above $40\ \mathrm{Nm}$)

B. Overstrain Failure (Light Green dots)Condition: These failures occur at the highest levels of Torque in the dataset (many above $60\ \mathrm{Nm}$) and are spread across various rotational speeds.

C. Heat Dissipation Failure (Teal dots) Condition: These failures mostly occur in the mid-range of Torque and mid-to-high range of Rotational Speed.

In [ ]:

In [ ]:

In [ ]:

Modeling

Evaluation

Recommendations

Next Steps

Thank you