Capstone Project Data Science Nanodegree

Project Definition

Project Overview

Predictive maintenance is a type of maintenance strategy that uses data and analytics to predict when equipment is likely to fail, so that maintenance can be scheduled in advance and downtime can be minimized. One of the main challenges of implementing predictive maintenance is getting access to the data that is needed to make accurate predictions.[1] This can be difficult for a number of reasons, such as the lack of sensors or monitoring systems on the equipment, the inability to collect and store the data in a usable format, or the lack of resources to analyze and interpret the data. Additionally, the data that is collected may be noisy or unreliable, making it difficult to draw accurate conclusions. Despite these challenges, predictive maintenance has the potential to significantly improve the efficiency and effectiveness of maintenance operations, and is worth considering for organizations that are looking to optimize their maintenance processes.[2]

Using synthetic data as a substitute for real data can be a useful approach in situations where it is difficult to obtain sufficient amounts of real data. Synthetic data is data that is generated artificially using algorithms and models, rather than being collected from real-world observations or experiments. It can be used to simulate a variety of different scenarios and conditions, and can be tailored to meet specific needs or requirements. One potential use of synthetic data is in the context of predictive maintenance, where it can be used to test and evaluate predictive models in situations where real data is hard to come by.[3] The data used in the analysis is synthetic data that reflects real predictive maintenance. It can be found on Kaggle[4].

Problem Statement

In this project we were given a dataset of equipment failures and non-failures, and build a model to classify each instance as a power failure or a non-power failure. The model should be able to accurately predict the type of failure based on features such as the equipment type, location, and operating conditions. The goal of this analysis is to identify patterns and trends in the data that may be indicative of power failures, in order to improve maintenance planning and minimize equipment downtime.

Metrics

In this analysis, we used machine data and atmospheric data as metrics to predict equipment failures and non-failures. The machine data included metrics such as rotational speed (rpm), torque (Nm), and tool wear (min), which provided information about the operating conditions and performance of the equipment. The atmospheric data included metrics such as air temperature (K) and process

---

[1] H. M. Hashemian, "State-of-the-Art Predictive Maintenance Techniques," in *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 1, pp. 226-236, Jan. 2011, doi: 10.1109/TIM.2010.2047662.
[2] Ran, Y., Zhou, X., Lin, P., Wen, Y., Deng, R., 2019. A Survey of Predictive Maintenance: Systems, Purposes and Approaches. https://doi.org/10.48550/ARXIV.1912.07383
[3] Carvalho, T.P., Soares, F.A.A.M.N., Vita, R., Francisco, R. da P., Basto, J.P., Alcalá, S.G.S., 2019. A systematic literature review of machine learning methods applied to predictive maintenance. Computers &amp; Industrial Engineering. https://doi.org/10.1016/j.cie.2019.106024
[4] Bansal, S. (2021) Machine Predictive Maintenance Classification, Kaggle. Available at: https://www.kaggle.com/datasets/shivamb/machine-predictive-maintenance-classification (Accessed: January 7, 2023).

temperature (K), which provided information about the environment in which the equipment was being used. These metrics were collected over time and used as input to a machine learning model, which was trained to classify each instance as a power failure or a non-power failure based on the values of these metrics. By analyzing patterns and trends in the data, the model was able to identify key features and characteristics that were indicative of power failures, and could be used to predict future failures with a high degree of accuracy.

Analysis

Data Exploration and Data Visualization

During the data exploration phase, we examined the features that we used as metrics in order to understand the relationships between them and the target variable (failure or non-failure). We looked at various statistics such as the mean, and plotted the data to visualize the distribution and trends. This process helped us to better understand the characteristics of the data and to identify any potential biases or issues that might affect the accuracy of our predictions.

In a scatter plot, we additionally show the distribution of two features and the occurrence of failures with the combination of the two features.

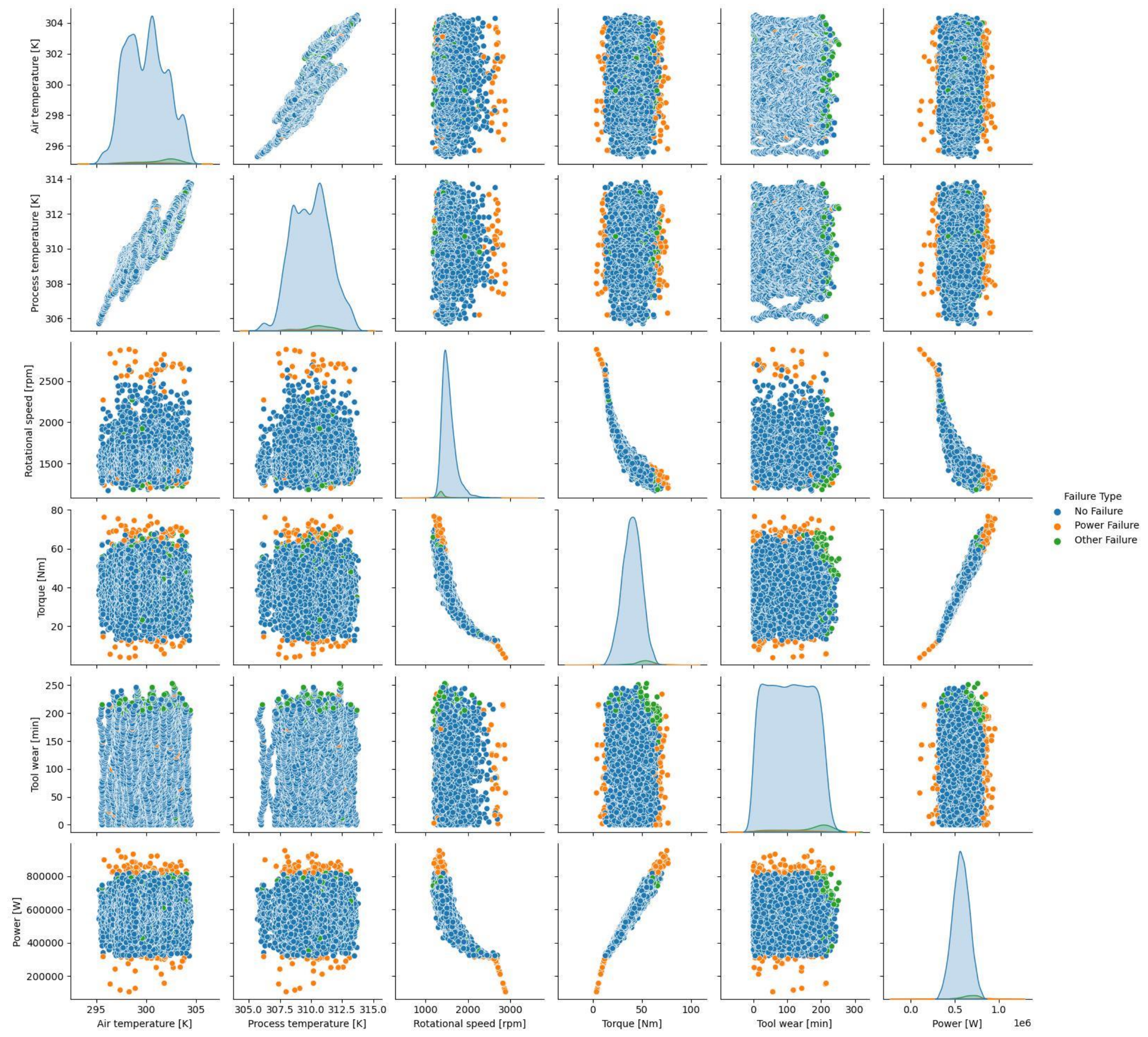
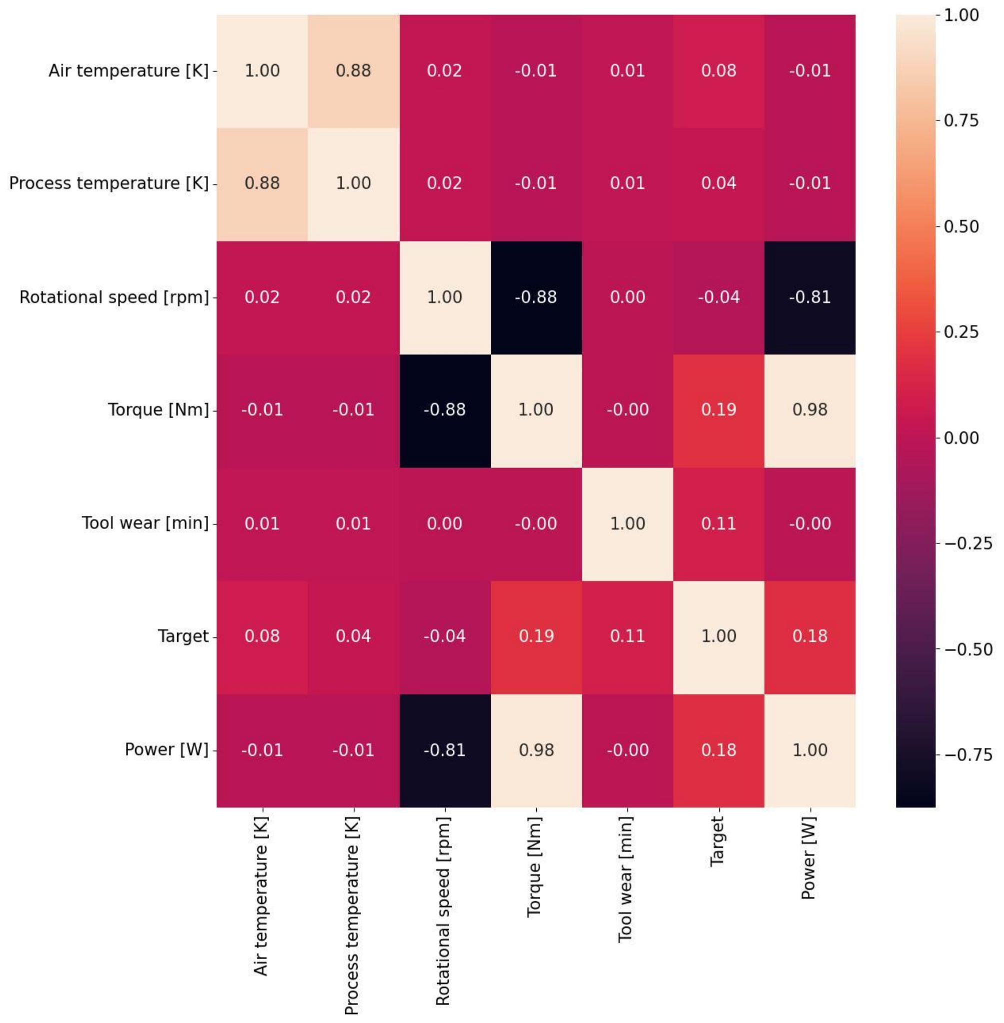


Figure 1: Distribution of paired features

Having a closer look at this distribution you can see that the power failure mainly occur in extreme situations. In the diagonal you can see the distribution of each feature unpaired. Looking at the power feature it is obvious that the power failure occurs below and above an specific watt.

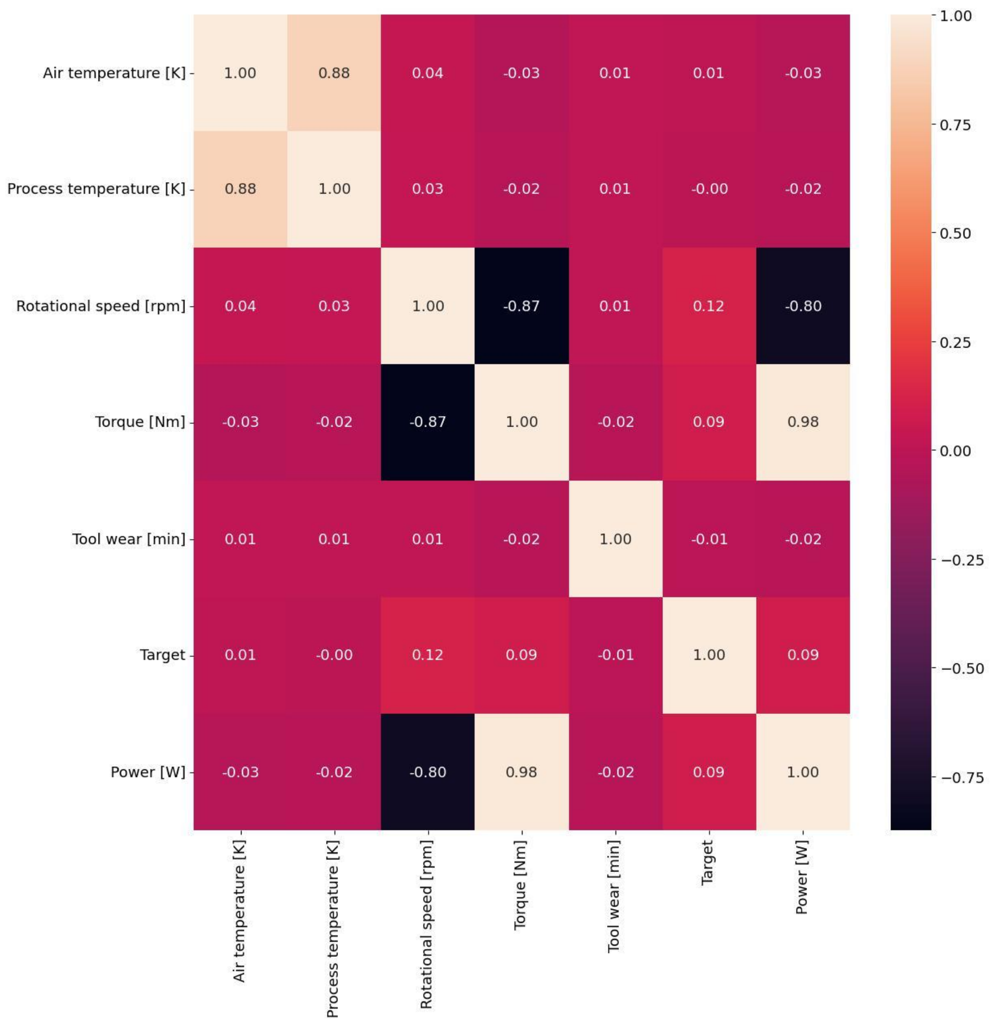
We also looked at the correlations between the features and the target variable, in order to identify any strong patterns or relationships that might be useful for predicting failures.



	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Power [W]
Air temperature [K]	1.00	0.88	0.02	-0.01	0.01	0.08	-0.01
Process temperature [K]	0.88	1.00	0.02	-0.01	0.01	0.04	-0.01
Rotational speed [rpm]	0.02	0.02	1.00	-0.88	0.00	-0.04	-0.81
Torque [Nm]	-0.01	-0.01	-0.88	1.00	-0.00	0.19	0.98
Tool wear [min]	0.01	0.01	0.00	-0.00	1.00	0.11	-0.00
Target	0.08	0.04	-0.04	0.19	0.11	1.00	0.18
Power [W]	-0.01	-0.01	-0.81	0.98	-0.00	0.18	1.00

Figure 2: Heatmap of the correlation of the different features

If we have a look at the target column we can see that almost all features correlate a little with the target column. It is impossible to identify the explaining feature for a failure as it is a mix from different features.



	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Target	Power [W]
Air temperature [K]	1.00	0.88	0.04	-0.03	0.01	0.01	-0.03
Process temperature [K]	0.88	1.00	0.03	-0.02	0.01	-0.00	-0.02
Rotational speed [rpm]	0.04	0.03	1.00	-0.87	0.01	0.12	-0.80
Torque [Nm]	-0.03	-0.02	-0.87	1.00	-0.02	0.09	0.98
Tool wear [min]	0.01	0.01	0.01	-0.02	1.00	-0.01	-0.02
Target	0.01	-0.00	0.12	0.09	-0.01	1.00	0.09
Power [W]	-0.03	-0.02	-0.80	0.98	-0.02	0.09	1.00

Figure 3: Heatmap of the correlation of the different features with target being a power failure or no failure at all

If we change the heatmap in a manner that the target column now only identifies whether we have a power failure or no failure at all (leaving out the other failures) we can see that the target now only correlates with the rotational speed, the torque, and the power.

Additionally, we used a histogram to explore the distribution of failure and no failure. The domination of no failures is clearly visible thereby. This already shows that increasing the size of failures would help the classification as we can get more patterns that show in which case the machine is likely to fail.

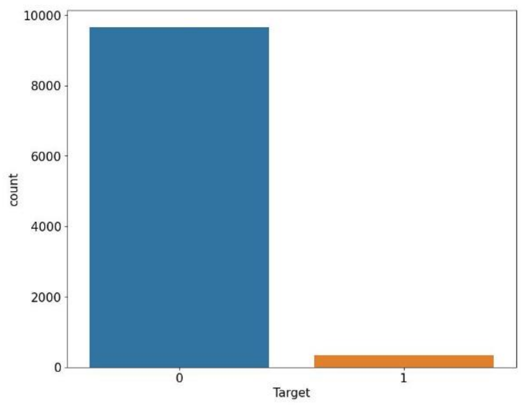


Figure 4: Histogram of count of failure (1) or no failure (0)

Methodology

Data Preprocessing

In the data preprocessing step, we cleaned the data by dropping irrelevant information that did not have an impact on the failure prediction process. For example, we may have removed columns such as product id and unique identifier, as these types of variables do not provide any useful information about the equipment or its operating conditions. These variables may have added noise to the data and could have negatively impacted the accuracy of our predictions. By removing them, we were able to focus on the features that were most relevant to the task at hand and reduce the complexity of the data. In general, data preprocessing is an important step in any machine learning project, as it helps to ensure that the data is in a usable format and is ready for further analysis and modeling. By cleaning and preparing the data in this way, we were able to more effectively and efficiently build and evaluate our predictive model.

In addition to the existing features, we also added a new feature to the data that was derived from the torque and rotational speed. This new feature can be calculated by applying a mathematical transformation to the torque and rotational speed values. Therefore, we took the product of the torque and rotational speed. By adding this soft measurement to the data, we were able to incorporate additional information that may have been useful for predicting equipment failures. This approach is known as feature engineering, and can be a powerful way to improve the performance of machine learning models by adding new features that capture relevant information that is not present in the raw data.

In addition, we also converted all failures that were not classified as power failures into a separate category called "other failures." This would have involved identifying all of the failures that were not related to power issues and grouping them into a single category. This approach would have simplified the problem by reducing the number of output classes that the model needed to predict. Instead of having to classify each failure as either a power failure or one of several different types of non-power failures, the model would only need to predict three classes: power failures, other failures and no failure. This can make the modeling process more straightforward and may improve the performance of the model. However, it is important to keep in mind that by grouping all non-power failures into a single category, we would have lost some of the granularity and detail in the data, which could potentially impact the usefulness of the predictions.

Implementation

To tackle the classification problem, we started off by transforming the explaining string variables (X) into a form that could be used as input to a machine learning model. To do this, we used an encoder to convert the string variables into binary variables. One way to do this is to use the OneHotEncoder from scikit-learn, which is a popular library for machine learning in Python. The OneHotEncoder takes a categorical feature as input and converts it into a binary matrix, with one column for each category and a value of 1 in the column corresponding to the category and a value of 0 in all other columns. This process is known as one-hot encoding, and is commonly used to represent categorical variables in machine learning models. By using the OneHotEncoder to transform the string variables, we were able to convert them into a format that the model could understand and use for prediction. This was an important step in the process of building and evaluating our predictive model.[5]

After transforming the string variables into a numerical form, we used a classifier to train a model to predict equipment failures. The classifier that we used was the LinearSVC, which stands for "Linear

[5] 6.3. Preprocessing Data (no date) scikit. Available at: https://scikit-learn.org/stable/modules/preprocessing.html#preprocessing-categorical-features (Accessed: January 8, 2023).

Support Vector Classification." This is a type of support vector machine (SVM) that is used for classification tasks. It works by finding the hyperplane in a high-dimensional space that maximally separates the classes. In our case, the classes were power failures, non-power failures and no failures.[6]

Refinement

To check if another classifier might be behaving better, we also used a Random Forest Classifier in addition to the LinearSVC. A Random Forest Classifier is an ensemble learning method that uses multiple decision trees to make predictions. It works by training multiple decision trees on different subsets of the data and then combining their predictions to make a final decision. This approach can be more robust and accurate than using a single decision tree, as it reduces the risk of overfitting and can improve the generalizability of the model. By using a Random Forest Classifier, we were able to compare its performance to that of the LinearSVC and see if it was able to achieve better results. This is a common practice in machine learning, as it allows us to compare the performance of different models and select the one that performs the best.[7]

By checking the confusion matrix of the classifier results, we were able to see how well the models were performing on the test data. A confusion matrix is a table that shows the number of true positive, true negative, false positive, and false negative predictions made by the model. It is often used to evaluate the accuracy of a classifier and to identify areas where the model might be struggling. In this case, we found that the Random Forest Classifier was not predicting anything other than no failures, while the LinearSVC was able to predict power failures, other failures, and no failures. This indicates that the LinearSVC may have been a more effective model for this task, as it was able to make a wider range of predictions and had a higher overall accuracy. By comparing the confusion matrices of the two models, we were able to gain insight into their relative performance and make a more informed decision about which one to use.

By just comparing the accuracy of both classifiers, it would not have been possible to indicate the better performing model, as both models had relatively high and comparable accuracy. Accuracy is a measure of how often the model correctly predicts the correct class, and is often used as a benchmark for evaluating the performance of a classifier.

Table 1: Accuracies of the used classifier

LinearCSV	RandomForest
0.96125	0.965

However, it is not the only metric that should be considered when comparing models. In some cases, a model with a high accuracy may not be the best choice, as it may be making many false positive or false negative predictions that are not acceptable for the given task.

[6] 1.4. Support Vector Machines (no date) scikit. Available at: https://scikit-learn.org/stable/modules/svm.html#svm-classification (Accessed: January 8, 2023).

[7] 1.11. ensemble methods (no date) scikit. Available at: https://scikit-learn.org/stable/modules/ensemble.html#forest (Accessed: January 8, 2023).

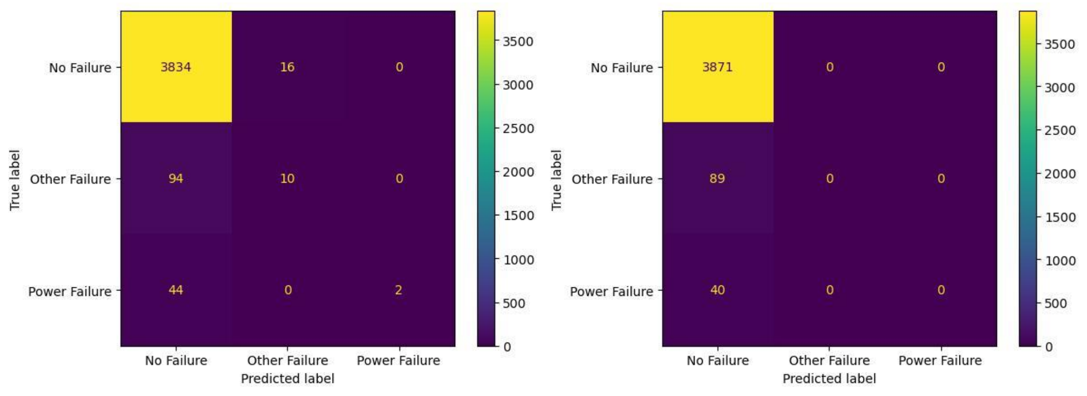


Figure 5: Comparing confusion matrix of LinearSCV (left) and RandomForest (right)

Comparing the confusion matrix of both classifiers, the LinearSVC performs better as it does predict power failure. Instead, the Random Forest does not predict any failure.

To improve the classification with the LinearSVC, we used GridSearch from scikit-learn to try different parameters for the model and find the best ones. GridSearch is a method for searching for the optimal set of hyperparameters for a machine learning model. It works by training the model with a range of different hyperparameter values and evaluating its performance on the test data. The hyperparameters of a model are the settings that control the behavior and performance of the model, and are typically set before training begins. By using GridSearch, we were able to systematically explore the space of possible hyperparameter values and select the ones that produced the best results. This is a common practice in machine learning, as it can help to improve the performance of the model and achieve better results on the test data. By using GridSearch to tune the hyperparameters of the LinearSVC, we were able to improve the accuracy and reliability of the model.[8]

Results

Model Evaluation and Validation

Based on the results, it appears that the LinearSVC was able to achieve an accuracy of around 96% on the test data. This is a relatively high accuracy, and suggests that the model was able to make accurate predictions in most cases. The model was also able to predict power failures, other failures and no failures, which is a good sign. However, the main problem is that the model often predicts no failure instead of any other type of failure. This is likely due to the fact that there are a large number of no failures in the dataset, which can make it difficult for the model to accurately predict the other types of failures. One way to address this issue would be to train the model with more failure data, in order to give it more examples to learn from and improve its ability to predict failures. This might involve gathering more data from real-world equipment failures, or synthesizing additional data using simulation or other methods. By increasing the amount of failure data that is available for training, it may be possible to improve the model's ability to predict failures and reduce the number of false negative predictions.

[8] 3.2. tuning the hyper-parameters of an estimator (no date) scikit. Available at: https://scikit-learn.org/stable/modules/grid_search.html#grid-search (Accessed: January 8, 2023).

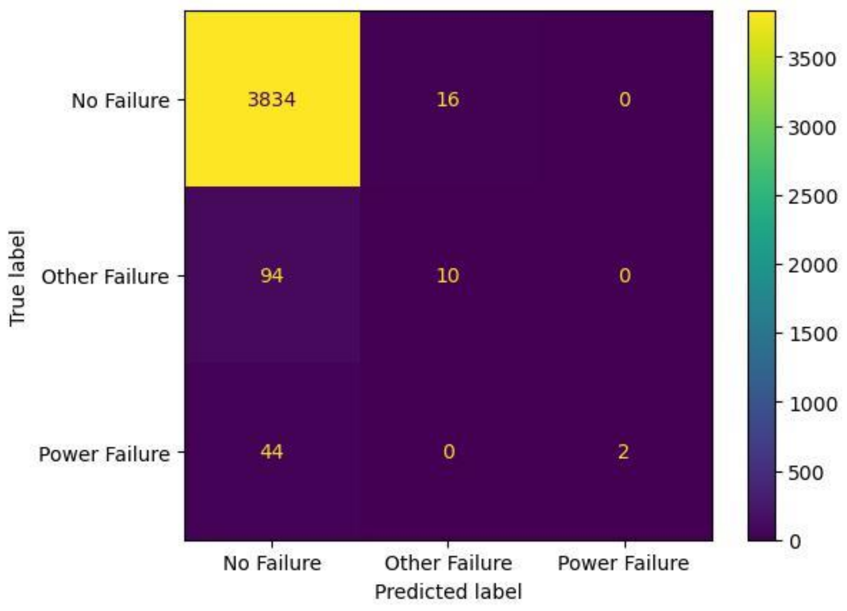


Figure 6: Confusion matrix of the LinearSVC

It appears that power failures are highly correlated with the power, which was calculated using the torque and rotational speed. Additionally, using a boxplot to visualize the occurrence of the failures and gain further insights into the data. A boxplot is a graphical representation of a set of data that shows the minimum, first quartile, median, third quartile, and maximum values, as well as any outliers. By examining the boxplot, you see how the failures were distributed across the data. By dividing the data into below and above the median we can identify with the boxplot when the machine is likely to fail according to power values.

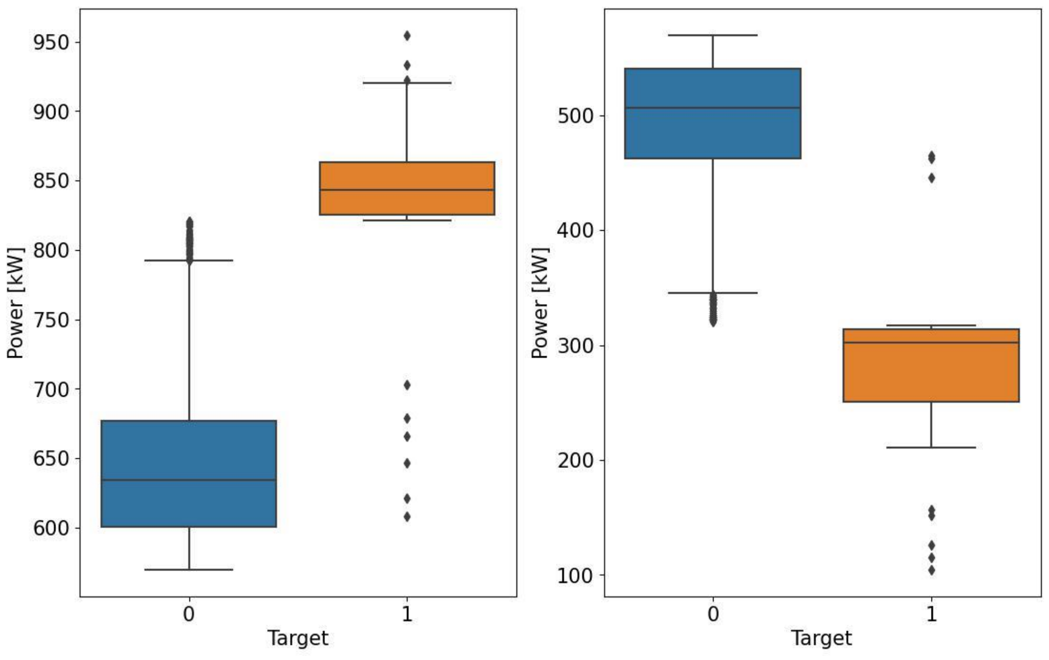


Figure 7: Boxplot of power over target (no failure = 0; power failure = 1)

In the boxplot you can see that we have a power failure above a certain power level and below a certain power level.

Justification
The model results that are seen in the confusion matrix can be improved as we have a high true negative rate for the classification of power failure. By just interpretating the boxplot we could get a similar result if we know from our engineer experts that this is the main reason for the failure. If we then do not get any other unexplainable failures just using the limit values for the power is an accurate solution to predicting the power failure.

Conclusion
Reflection
Overall, the problem that we extensively discussed above was focused on predicting equipment failures and classifying them as either power failures or non-power failures. In order to solve this problem, we used machine learning techniques to build a predictive model that could analyze data on equipment failures and identify patterns and trends that were indicative of power failures. We started by cleaning and preprocessing the data, and then used an encoder to transform the string variables into a numerical form that could be used as input to a machine learning model. We then trained and evaluated two different classifiers: a LinearSVC and a Random Forest Classifier. By comparing the confusion matrices of the two models, we were able to see that the LinearSVC was performing better, and decided to use it as our final model. We then used GridSearch to tune the hyperparameters of the LinearSVC and improve its accuracy. Finally, we analyzed the correlations between the power and the failure data, and used a boxplot to visualize the occurrence of the failures. The main difficulty is the classification of power failure. More data, explicitly power failure data, could be helpful to improve the prediction of the failure.

Improvement
There are several possible improvements that could be made in future work on this problem:

- Including class weights for the classification. In a manner that predicting power failure true true is more important than predicting no failure true true.
- Gather more data: One potential limitation of the current approach is the amount of data that is available for training the model. By collecting more data from real-world equipment failures, it may be possible to improve the accuracy and generalizability of the model. Especially collecting more failure data can result in a better prediction of failures of the model.
- Explore different modeling approaches: There are many different machine learning algorithms and techniques that could be used to solve this problem. By exploring different approaches, it may be possible to find a model that performs better than the LinearSVC.
- Incorporate additional features: The current model is based on a limited set of features. By adding additional features that capture more information about the equipment and its operating conditions, it may be possible to improve the performance of the model.
- Incorporate domain knowledge: By working closely with experts in the field of equipment maintenance, it may be possible to incorporate additional domain knowledge into the model and improve its performance.