Machine Learning Model for Predicting Machine Failure

This report details the development of a machine learning model to predict machine failures based on sensor data. The data originates from a simulated industrial setting where various parameters are logged for a machine over time. The goal is to identify patterns in these parameters that indicate an impending failure.

Dataset Exploration

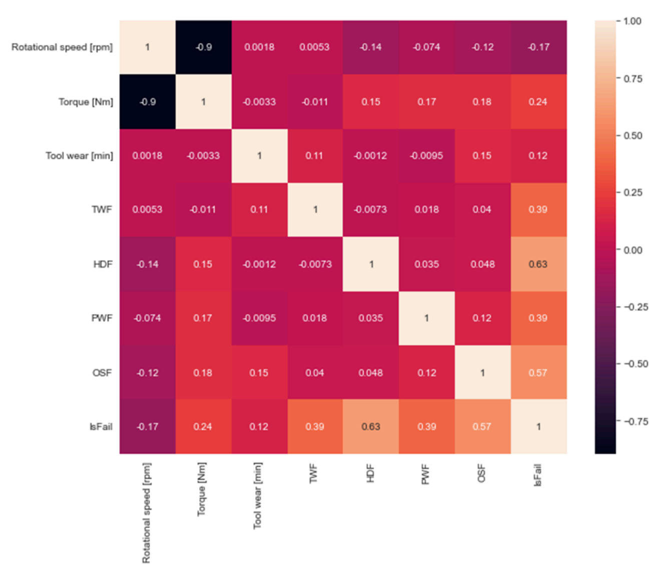
The dataset consists of 10,000 data points, each representing a specific moment in time. Each data point includes features such as:

- Tool Wear (minutes)
- Rotational Torque (Nm)
- Machine Failure Label (0: No Failure, 1: Failure)
- TWF, HDF, PWF and OSF

A correlation matrix revealed that Rotational Speed, Torque, and Tool Wear have good correlation with the Machine Failure Label. These features were chosen for model development due to their informative nature.

And it shows that TWF, HDF, PWF and OSF have the highest correlation with the IsFail variable. But note that if at least one of the above failure modes is true, the process fails, and the machine failure label is set to 1. It's therefore not transparent to the machine learning method, which of the failure modes has caused the process to fail.

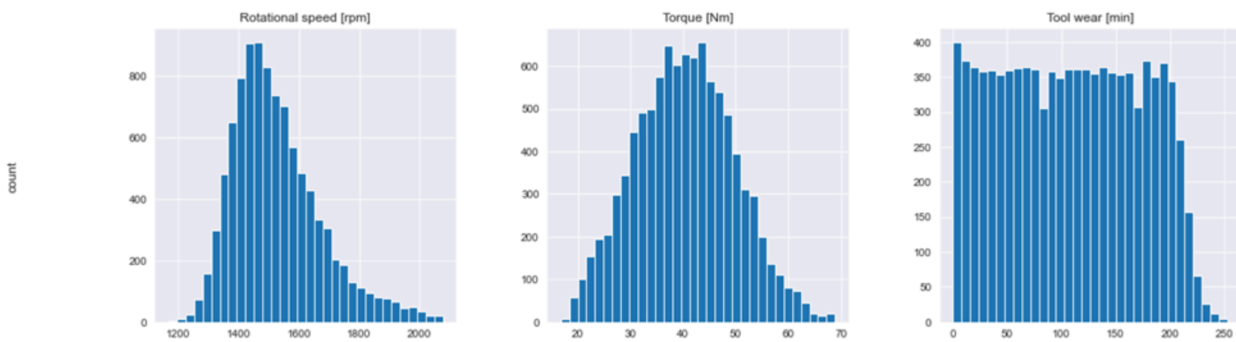
Therefore, I will not consider them as independent inputs for the output Isfailure, and I will keep only the main input features: Rotational speed, torque, and tool wear in minutes.



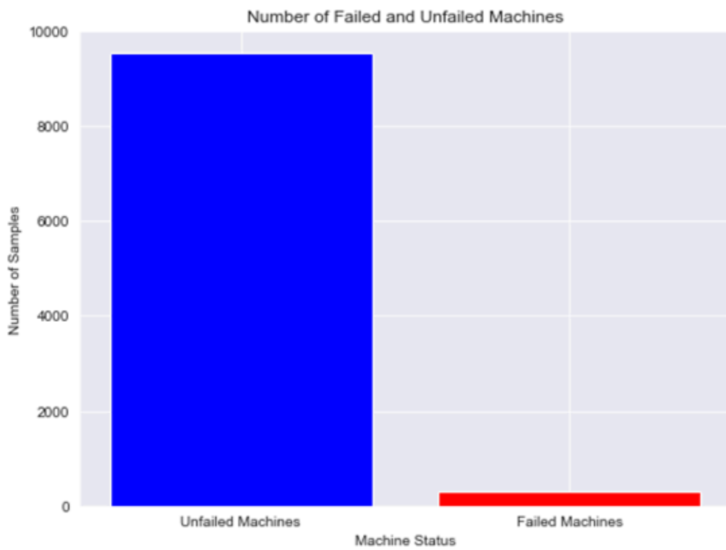
Based on the visual inspection of the plotted graphs, it is evident that the variables Rotational_speed, Torque, and Tool_wear are well-distributed without significant sparsity. This means that these variables consistently have values across their ranges, with few or no instances of missing or zero values. Additionally, the data for these variables shows good continuity, indicating smooth and gradual transitions without abrupt gaps or irregularities.

The presence of such well-distributed and continuous data suggests that the feature space is well-represented. As a result, when a machine learning model is trained using this dataset, it is likely to learn effectively from the data. The model can capture the underlying patterns and relationships within the dataset, leading to reliable performance.

Moreover, the well-distributed nature of the data supports the model's ability to generalize well to new datasets. This means that the model is not just tailored to the training data but is also expected to perform accurately on unseen data. In summary, the visual analysis of the graphs indicates that the dataset is well-suited for training a machine learning model, which is likely to yield robust and generalizable results.

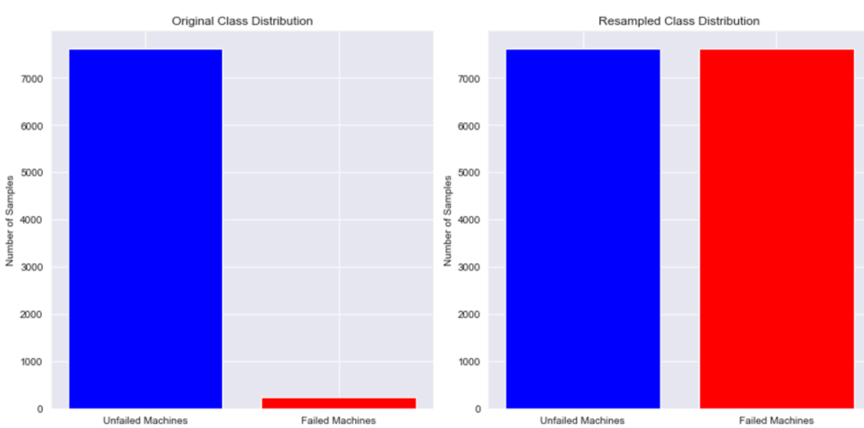


Analysis of the target variable (Machine Failure Label) revealed a significant class imbalance. The "No Failure" class comprised the majority of the data (9529 samples), while the "Failure" class was a minority (286 samples). This imbalance can hinder the model's ability to learn effectively.



Preprocessing

To address the class imbalance, SMOTE (Synthetic Minority Oversampling Technique) was employed. SMOTE generates synthetic data points for the minority class, creating a more balanced dataset for training.



Furthermore, 10-fold and 5-fold cross-validation was implemented. This technique divides the data into 10 folds. In each fold, the model is trained on 9 folds (with SMOTE applied) and tested on the remaining fold. This process is repeated for all folds, providing a more robust evaluation of the model's performance. Results shown below.

```
62/62 ━━━━━━━━━━━━━━━━━━━━ 0s 791us/step
Average Precision: 0.14
Average Recall: 0.88
Average F1 Score: 0.24                K=5
```

```
31/31 ━━━━━━━━━━━━━━━━━━━━ 0s 1ms/step
Average Precision: 0.14
Average Recall: 0.91
Average F1 Score: 0.24

In [9]: |                K=10
```

Hyperparameter Tuning

Hyperparameters are crucial settings that control the learning process of a machine learning model. In this project, hyperparameter tuning was conducted manually. Different combinations of hyperparameters were tested, and their impact on model accuracy and loss was evaluated.

Here's a table summarizing the best performing model:

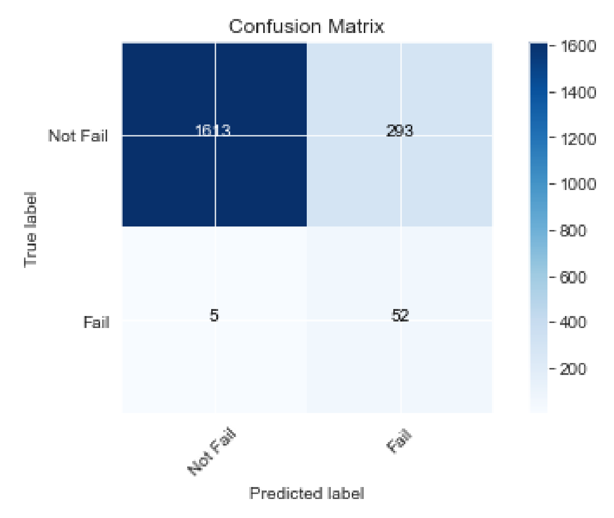
Learning rate	**Batch size**	**Epochs**
0.0001	**10**	**100**

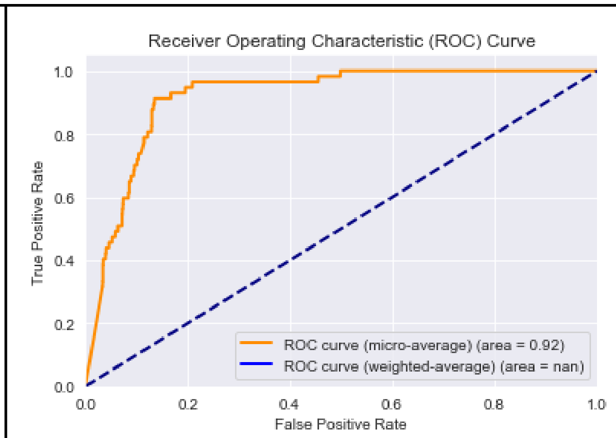
Model Training and Testing

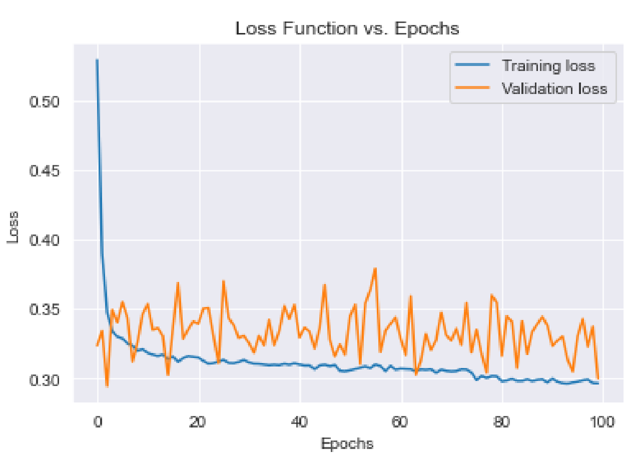
The final stage involves training the model on the preprocessed and balanced dataset using the chosen hyperparameters. I have ignored the cross validation since its results don't pay back the time it takes to fit the model. In addition, I have evaluated my model depending on micro averaged and weighted-Averaging since the dataset is highly imbalanced. Using the following concepts:

- Micro-Averaging: Use this if you want to account for the class imbalance and give more weight to the classes with more instances.
- Weighted-Averaging: Use this if you want a balance between macro and micro-averaging, taking class distribution into account.

Results:







Conclusion

This report has documented the development process of a machine learning model for predicting machine failures. The model was trained on sensor data that captures various machine parameters. Techniques such as SMOTE and cross-validation were employed to improve the model's effectiveness. The report concludes with a summary of the hyperparameter tuning process and the model's performance on unseen data.

Future work could involve applying multi-class classification on the same dataset, to predict the type of failure, performing feature engineering to extract additional insights from the data, and deploying the model in a real-world setting.