UNIVERSITÀ DI PISA

Data Mining and Machine Learning

Machine Failure Classifier

Francesco De Vita

Index

1. Introduction

This project aims to apply data analysis and machine learning techniques for fault prediction in industrial production, using a real dataset from a metal component manufacturing process. The dataset contains information on various physical and chemical parameters of the process, such as temperature, rotation speed, torque, tool wear and machine failure types. The goal of the project is to train a machine learning model to detect damage, classifying it into different types of failures.

A graphical user interface application was also developed to test the trained model.

2. Dataset

The dataset we are using is called 'ai4i2020' and comes from an industrial production process of metal components.
The dataset contains 10 000 rows and 14 columns, each of which represents a feature of the process, such as air temperature, rotational speed, torque, tool wear and machine failure types. Our goal is to use these features to predict failures and classify them into different types, such as tool wear, overheating, overvoltage, overload and random failures. The dataset is synthetic, but reflects the real data that is encountered in industry.

1. UID: unique identifier ranging from 1 to 10000
2. 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
3. type: just the product type L, M or H from column 2
4. air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K
5. 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.
6. rotational speed [rpm]: calculated from a power of 2860 W, overlaid with a normally distributed noise
7. torque [Nm]: torque values are normally distributed around 40 Nm with a SD = 10 Nm and no negative values.
8. tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process.
9. 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

1. tool wear failure (TWF): the tool will be replaced of 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).
2. 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 tools rotational speed is below 1380 rpm. This is the case for 115 data points.

3. 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.

4. 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.

5. 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.

	UDI	Product ID	Type	Air temperature [K]	Process temperature [K]	Rotational speed [rpm]	Torque [Nm]	Tool wear [min]	Machine failure	TWF	HDF	PWF	OSF	RNF
0	1	M14860	M	298.1	308.6	1551	42.8	0	0	0	0	0	0	0
1	2	L47181	L	298.2	308.7	1408	46.3	3	0	0	0	0	0	0
2	3	L47182	L	298.1	308.5	1498	49.4	5	0	0	0	0	0	0
3	4	L47183	L	298.2	308.6	1433	39.5	7	0	0	0	0	0	0
4	5	L47184	L	298.2	308.7	1408	40.0	9	0	0	0	0	0	0

1 Dataset Sample

3. Data Pre-Processing

3.1 Attribute Creation

We have created attributes to capture some information that will help us in the prediction, these 3 elements are the power, tool_torque and delta_temperature

power: is the power absorbed by the machine during the process, calculated as the product of the rotational speed and the torque. Usefull to predict power failure (PWF).

tool_torque: Is the product between the input torque and the output torque. Usefull to predict overstrain failure (OSF).

delta_temperature: is the temperature difference between the air and the metal component, calculated as the air temperature minus the process temperature. Usefull to predict heat dissipation failure (HDF).

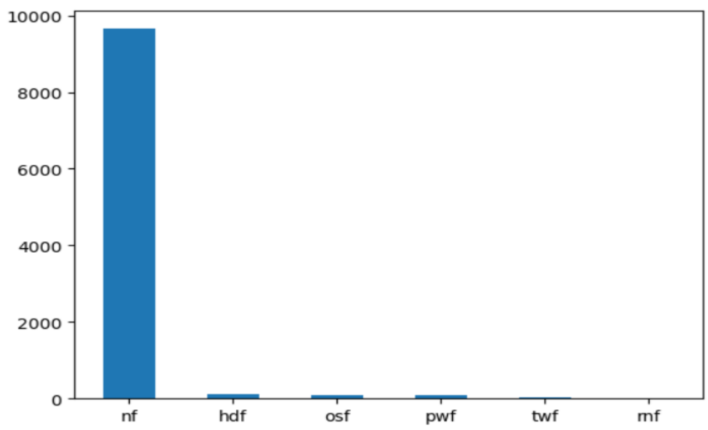
```
df['power']=(df['rotational_speed']/60)*(6.28)*df['torque']
df['tool_torque']= df['tool_wear']*df['torque']
df['delta_temperature']= df['air_temperature']-df['process_temperature']
```

2 Attribute Creation

The features of this dataset that represent the fails have been grouped into a single feature. The rows that have more failures will be duplicated and each duplicate will have a different failure.

The dataset is very unbalanced, because in the field of predictive maintenance it is difficult to find data that reflect the failure of a machine, about 9600 rows belong to the class ‘nf’ no failure, the remaining ones are distributed with a very low number of samples in the other 5 classes. For this reason it will be necessary to perform an oversampling with SMOTE

nf	9652
[hdf]	106
[pwf]	80
[osf]	78
[twf]	42
[rnf]	18
[pwf, osf]	11
[hdf, osf]	6
[hdf, pwf]	3
[twf, osf]	2
[twf, rnf]	1
[twf, pwf, osf]	1

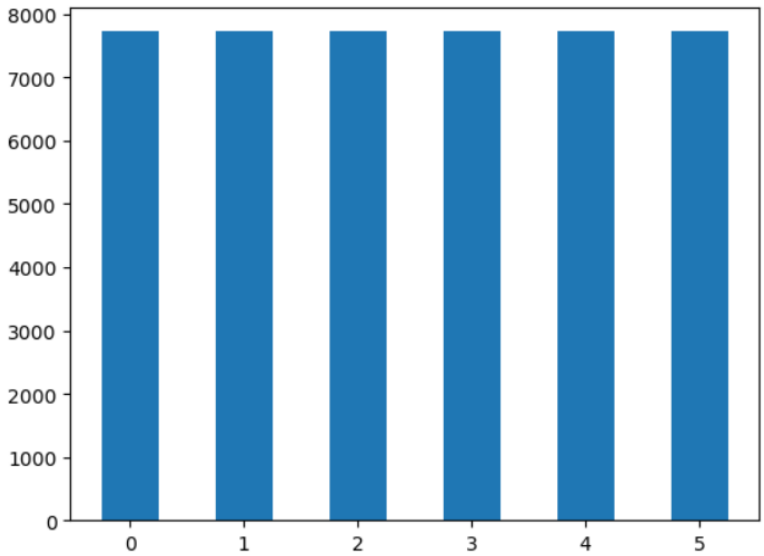


3 Class Distribution

3.2 Train-Test split

The first step after removing the null values and encode the features the dataset was divided into training and test sets. This prevented any information from the preprocessing steps from leaking into the test set. The test set size was set to 0.3, meaning it contained 30% of the data. The Stratify option was used to keep the same proportion of classes in both sets.

Than we apply the SMOTE Oversampling to the Training set



4 SMOTE Oversampling

3.3 Outlier Detection

Looking at the distributions and boxplots for each attribute for each class, we can state that there are no outliers, values that deviate considerably from the mean or the median.

3.4 Feature Selection

This operation consists of selecting the most relevant features for the classification model, eliminating those that do not provide useful information or that are correlated with each other. To do this, the mutual information criterion was used, in particular, the “mutual_info_classif” method was applied, which calculates the mutual information between each feature and the class variable, returning a value between 0 and 1.

$$I(C, fi) = \sum_{c \in CC} \sum_{f_i \in FF} p(c, fi) \cdot \log \frac{p(c, fi)}{p(c)p(fi)}$$

$$NI(f_i, f_s) = \frac{I(f_i; f_s)}{min\{H(f_i), H(f_s)\}}$$

$$G = I(C, f_i) - \frac{1}{S} \sum_{f_s \in S} NI(f_i; f_s)$$

5 Mutual Information

I decided to eliminate the features that had a mutual information very close to zero, the features eliminated are air_temperature, process_temperature and type.

3.5 Normalization

Since the data do not have any outliers, we can apply a MinMax normalization to the features, in order to reduce the disparity between the values of some of them.

```
from sklearn.preprocessing import MinMaxScaler
sc = MinMaxScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

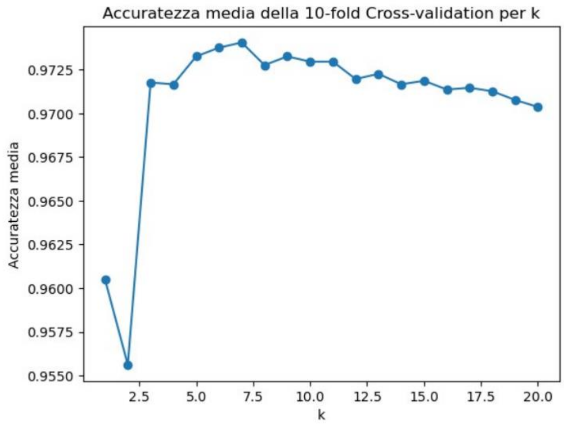
4. Evaluation

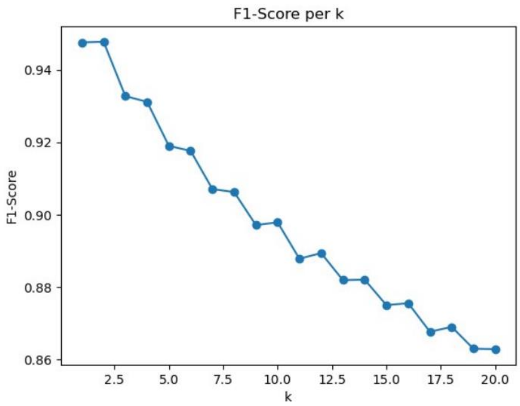
To evaluate the performance of the classifiers, the accuracy and F1-Score metrics are used. The F1-Score is especially important because the dataset has a strong imbalance between the classes, and the accuracy alone is not a reliable indicator of the quality of the classifier.

The Classes from 0 to 5 in the images represents resepectively ['hdf' , 'nf', 'osf', 'pwf', 'rnf', 'twf']

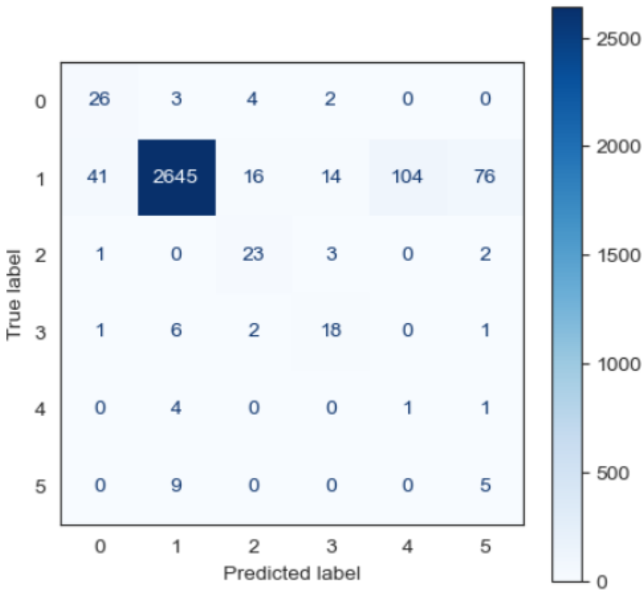
KNN

The best k value for KNN is 3 because it’s the right trade between Accuracy and F1-Score





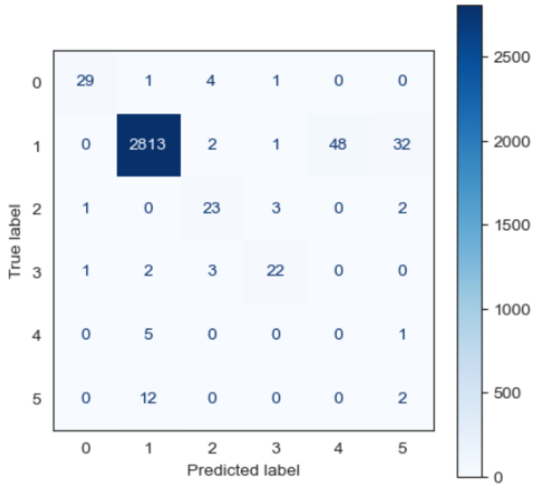
	precision	recall	f1-score	support
0	0.38	0.74	0.50	35
1	0.99	0.91	0.95	2896
2	0.51	0.79	0.62	29
3	0.49	0.64	0.55	28
4	0.01	0.17	0.02	6
5	0.06	0.36	0.10	14
accuracy			0.90	3008
macro avg	0.41	0.60	0.46	3008
weighted avg	0.97	0.90	0.93	3008



Decision Tree

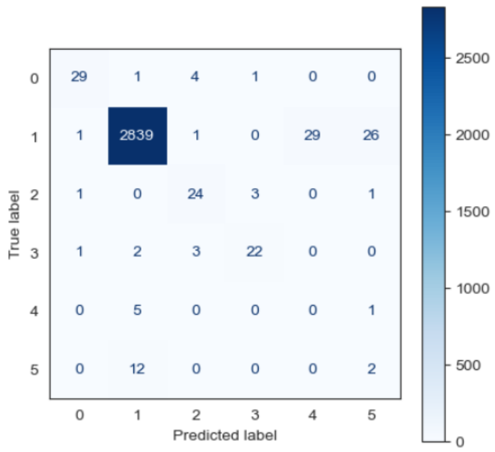
Classification report:

	precision	recall	f1-score	support
0	0.94	0.83	0.88	35
1	0.99	0.97	0.98	2896
2	0.72	0.79	0.75	29
3	0.81	0.79	0.80	28
4	0.00	0.00	0.00	6
5	0.05	0.14	0.08	14
accuracy			0.96	3008
macro avg	0.59	0.59	0.58	3008
weighted avg	0.98	0.96	0.97	3008



RandomForest

	precision	recall	f1-score	support
0	0.91	0.83	0.87	35
1	0.99	0.98	0.99	2896
2	0.75	0.83	0.79	29
3	0.85	0.79	0.81	28
4	0.00	0.00	0.00	6
5	0.07	0.14	0.09	14
accuracy			0.97	3008
macro avg	0.59	0.59	0.59	3008
weighted avg	0.98	0.97	0.98	3008

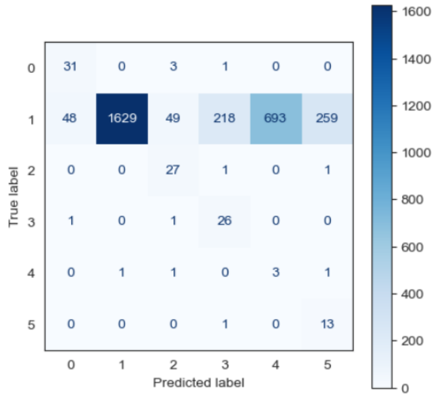


Bayesian Classifier

```
              precision    recall  f1-score   support

           0       0.39      0.89      0.54        35
           1       1.00      0.56      0.72      2896
           2       0.33      0.93      0.49        29
           3       0.11      0.93      0.19        28
           4       0.00      0.50      0.01         6
           5       0.05      0.93      0.09        14

    accuracy                           0.57      3008
   macro avg       0.31      0.79      0.34      3008
weighted avg       0.97      0.57      0.71      3008
```



4.1 Results

From the results of the classification report and the confusion matrix, it emerges that the most performing classifier is the Random Forest, which achieved the highest values in all the metrics. It was predictable that all the models would encounter difficulties in classifying class 4 and 5 (‘rnf’ and ‘twf’), as they are not failures determined by the attributes, but random events or dependent on the wear of the machine (rnf is a sudden failure of a machine, while twf is a failure caused by the failure to renew a worn-out machine).

The misclassification of some datapoints could be attributed to the fact that they were classified with more than one error in the initial dataset.

	hdf	nf	osf	pwf	rnf	twf
KNN	0.50	0.95	0.62	0.55	0.02	0.10
Decision Tree	0.88	0.98	0.75	0.80	0.00	0.08
Random Forest	0.87	0.99	0.79	0.81	0.00	0.09
Bayesian Classifier	0.54	0.72	0.49	0.19	0.01	0.09

F1-Score for each Class

	KNN	Decision Tree	Random Forest	Bayesian Classifier
F1-Score	0.93	0.97	0.98	0.71

Average F1-Score

5. Comparison with other Studies

Another study examines this classification problem using a Bagged Decision Tree model. It trained 15 trees, the features of each tree were randomly selected from the 6 available in the dataset. Each tree uses 4 features out of 6, following the pattern of Table II. The selection criterion was to maximize the sum of the predictive importance of the features (the higher the importance, the more likely the choice).

TABLE II. FEATURES USED FOR THE TRAINING OF THE EXPLAINABLE DECISION TREES. BLANK FIELD INDICATE FEATURE WAS NOT PROVIDED, WHEREAS 0 INDICATES THAT PROVIDED FEATURE WAS NOT USED. A 1 INDICATES THAT THE FEATURE IS USED IN THE DECISION TREE.

tree num.	air temp.	proc. temp.	rot. speed	torq.	type	tool wear
1			1	1	0	1
2		0		1	0	1
3		0	1		1	1
4		0	1	1		0
5		0	1	1	0	
6	0			1	0	1
7	1		1		0	1
8	0		1	1		0
9	1		1	1	0	
10	0	0			0	0
11	0	0		1		1
12	0	0		1	1	
13	1	1	1			0
14	1	1	1		0	
15	1	1	1	1		
∑	5	3	10	10	2	6

The only way to compare this study is through the F1-Score, because the evaluation of the classifier was done by the authors of the paper on a sample of 20 Samples.

TABLE I. CONFUSION MATRIX OF THE BAGGED TREES ENSEMBLE CLASSIFIER USING 5-FOLD CROSS VALIDATION.

		true class	
		failure	operation
predicted class	failure	294 (86.7 %)	45 (13.3 %)
	operation	121 (1.3 %)	9,540 (98.7 %)

$$Precision = \frac{9540}{9540 + 121} = 0.98$$

$$Recall = \frac{9540}{9540 + 45} = 0.99$$

$$F1 - Score = \frac{2 * 0.98 + 0.99}{0.98 + 0.99} = 0.98$$

The results are quite similar to our model but in this study they used a simpler model, than as the principle of Occam’s razor suggests, it is preferable to use simpler models with the same accuracy, because more complex models can tend to overfit better to the training data.

Another thing we can notice is that the classifier proposed in the paper struggles to recognize TWF failures, as tuple are label in a “random” way.

TABLE V. SCORED EXPLAINATION OF DECISION TREES FOR SELECTED DATAPOINTS (CF. TABLE IV). SCORES (SCR) RANGE FROM ++ (VERY), + (PARTIALLY), − (LIMITED), TO − − (NO) USEFUL EXPLAINATION. THE SELECTED TREE IS INDICATED BY BOLD TYPE.

UID	Mode	BTC	Trees	Explanation	Scr
2672	TWF	0		none	n/a
3866	TWF	1		none	− −
6341	TWF	0		none	n/a
8358	TWF	0		none	n/a
9019	TWF	0		none	n/a
3237	HDF	0		none	n/a
4079	HDF	1		none	− −
4174	HDF	1		none	− −
4327	HDF	1	13-**15**	torque < 65 Nm rotSpeed < 1380 rpm airTemp < 301.5 K procTemp < 310.5 K	+ +
4502	HDF	1			+ +
464	PWF	1	1-**8**,9, 11-15	torque < 13.2 Nm	+
1493	PWF	1		none	− −
3001	PWF	1	1,2,4, 5,6,**8**, 9,11, 12,15	torque > 65 Nm rotSpeed > 1229 rpm	+ +
7537	PWF	1	1-5, 6-**8**, 9-15	torque < 13.2 Nm	+
8583	PWF	1	1,2, 4-6,**8**, 9,11, 12,15	torque > 65 Nm rotSpeed > 1229 rpm	+ +
250	OSF	1	2,6, **11**	torque > 53.6 Nm tool wear > 194.5 min	+ +
3020	OSF	1	2,3, 6,**11**		+ +
5400	OSF	1	2,6, **11**		+ +
7592	OSF	1	2,6, **11**		+ +
9660	OSF	1	2,3, 6,**11**		+ +

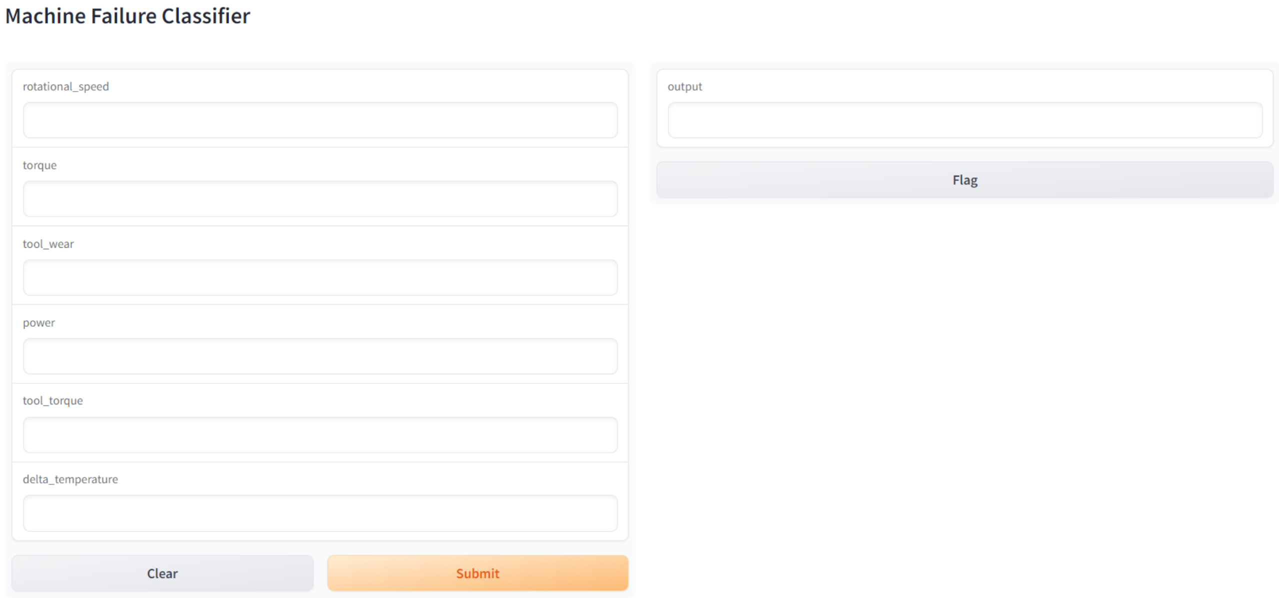
A problem that can be observed in the previous Table is that many samples do not have an explanation associated, because the decision trees fail to classify them correctly. To overcome this problem, a normalization technique based on the Z-Score was applied, which takes into account the standard deviation and the expected value of the variables. In this way, the two attributes that have the highest Z-Score in absolute value were selected, and used as an explanation for the samples. This solution does not guarantee a high accuracy of the explanations, but it can be helpful to help experts with some hints to identify the cause of the problem.

UID	Mode	air temp.	proc. temp.	rot. speed	torq.	tool wear
2672	TWF	299,7	309,3	1399	41,9	221
		-0,15	-0,47	**-0,78**	0,19	**1,78**
3237	HDF	300,8	309,4	1342	62,4	113
		0,40	-0,41	**-1,10**	**2,25**	0,08
464	PWF	297,4	308,7	2874	4,2	118
		-1,30	-0,88	**7,45**	**-3,59**	0,16
250	OSF	298	308,3	1405	56,2	218
		-1,00	-1,15	-0,75	**1,63**	**1,73**

UID	Mode	BTC	Explanation	Scr
2672	TWF	0	high tool wear of 221 mins low rot. speed of 1399 rpm	+
3866	TWF	1	high tool wear of 228 mins high air temp. of 302.6 K	+
6341	TWF	0	high tool wear of 210 mins low rot. speed of 1397 rpm	+
8358	TWF	0	high tool wear of 210 mins low rot. speed of 1397 rpm	+
9019	TWF	0	high tool wear of 217 mins low air temp. of 297.3 K	+
3237	HDF	0	high torque of 62.8 Nm low rot. speed of 1342 rpm	+
4079	HDF	1	high torque of 62.8 Nm low rot. speed of 1294 rpm	+
4174	HDF	1	high air temp. of 302.2 K low rot. speed of 1346 rpm	+ +
4327	HDF	1	high torque of 55.8 Nm low rot. speed of 1362 rpm	+
4502	HDF	1	high torque of 54.0 Nm low rot. speed of 1307 rpm	+ +
464	PWF	1	high rot. speed of 2874 rpm low torque of 4.2 Nm	+
1493	PWF	1	high torque of 58.5 Nm low air temp. of 298 K	+
3001	PWF	1	high torque of 72.8 Nm low rot. speed of 1324 rpm	+
7537	PWF	1	high rot. speed of 2579 rpm low torque of 12.5 Nm	+
8583	PWF	1	high torque of 72.8 Nm low proc. temp. of 308.1 K	+
250	OSF	1	high tool wear of 218 mins high torque of 56.2 Nm	+ +
3020	OSF	1	high tool wear of 207 mins high torque of 54.2 Nm	+ +
5400	OSF	1	high tool wear of 218 mins high air temp. of 302.8 K	+
7592	OSF	1	high torque of 61.3 Nm high tool wear of 202 mins	+ +
9660	OSF	1	high torque of 61.9 Nm high tool wear of 216 mins	+ +

6. User Interface

To show the practical use of the model, we created a simple user interface for the application, using the Gradio framework. In this way, you can easily interact with the model, entering the input data and observing the class predicted by the model in real time.



7. Future Improvements

Moreover, as a possible improvement, it would be interesting to add explanations generated by the model for each sample, based on the most relevant factors. These explanations could help to understand the reasoning of the model and to evaluate its reliability.

To make the model more effective, you could think of applying it to real industrial systems, which need to monitor the state of the machines and prevent failures. In this way, the model could provide the operators with useful information for the maintenance and diagnosis of the problems, improving the safety and efficiency of the machines. You could also integrate the model with other sources of data, such as sensors to acquire new data and make the model more robust and reliable, thus increasing its predictive ability and its accuracy.