ELECTRICAL ENGINEERING

EE257 MACHINE LEARNING

PROJECT

AI4I 2020 PREDICTIVE MAINTENANCE

GUIDANCE:

BIRSEN SIRKECI

By:

Niharika Nagaraja Gupta

SJSU ID: 016023384

DATASET DESCRIPTION

The AI4I 2020 predictive maintenance dataset is taken from the taken from the UCI Machine Learning repository. It is a synthetic dataset that mimics the real predictive dataset encountered in industry.

The dataset consists of 10,000 data points with 14 input features. The description of the following input features is as follows:

1. UID: unique identifier that ranges from 1 to 10000

2. Product ID: Contains a combination of alphanumeric characters, the first character is an alphabet(M,H,L) followed by a five digit number. The alphanumeric character gives details about the variant-specific serial number

3. Type: Indicates the product quality variants. It is encoded using 3 letters L(50% of all products),M(30%) or H(20%).

4. Air Temperature[K}: This column of the dataset is generated using a random walk process normalized to a standard deviation of 2K around 300K.

5.Process Temperature[K]: This part of the dataset is produced by making use of random walk process normalized to a standard deviation of 2K and then added to the air temperature plus 10K.

6.Rotational Speed[rpm]: Rotational speed is 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 Ïf = 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. and 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:

9. Tool Wear Failure (TWF): the tool will be replaced of fail at a randomly selected tool wear time between 200 â€“ 240 mins (120 times in our dataset). At this point in time, the tool is replaced 69 times, and fails 51 times (randomly assigned).

10. 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 toolâ€™s rotational speed is below 1380 rpm. This is the case for 115 data points.

11. 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.

12. 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.

13. 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.

14. 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.

First few columns of the dataset is shown below:

```
In [3]: data.head()
```

Out[3]:

	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

Output Variable:

According to the dataset, ‘Machine Failure’ is considered as output and is set to True(1) if at least any of the five independent failure modes become true(1). However, there are inconsistencies in the dataset wherein the ‘Machine Failure’ is set to False (0) even if one of the independent failure modes are true.

```
In [11]: data[points == False]
```

Out[11]:

	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
1221	1222	M16081	M	297.0	308.3	1399	46.4	132	0	0	0	0	0	1
1302	1303	L48482	L	298.6	309.8	1505	45.7	144	0	0	0	0	0	1
1437	1438	H30851	H	298.8	309.9	1439	45.2	40	1	0	0	0	0	0
1748	1749	H31162	H	298.4	307.7	1626	31.1	166	0	0	0	0	0	1
2072	2073	L49252	L	299.6	309.5	1570	35.5	189	0	0	0	0	0	1
2559	2560	L49739	L	299.3	309.0	1447	50.4	140	0	0	0	0	0	1
2749	2750	M17609	M	299.7	309.2	1685	28.9	179	1	0	0	0	0	0
3065	3066	M17925	M	300.1	309.2	1687	27.7	95	0	0	0	0	0	1
3452	3453	H32866	H	301.6	310.5	1602	32.3	2	0	0	0	0	0	1
4044	4045	M18904	M	301.9	310.9	1419	47.7	20	1	0	0	0	0	0

Therefore, another column ‘MF’ is created which performs OR operation of the five independent failure modes. The ‘MF’ is considered as output.

Input Variables:

The rest of the variables are considered as inputs. The input columns are ‘UDI’, ‘Type’, ‘Air Temperature’, ‘Process Temperature’, ‘Rotational Speed’, ‘Torque’, ‘Tool Wear’, ‘TWF’, ‘HDF’, ‘PWF’, ‘OSF’, ‘RNF’.

Problem Statement: The goal is to predict whether a machine will fail or not based on the inputs provided in the predictive maintenance dataset. The problem is considered as a binary classification problem.

Datatypes of all the variables is shown below:

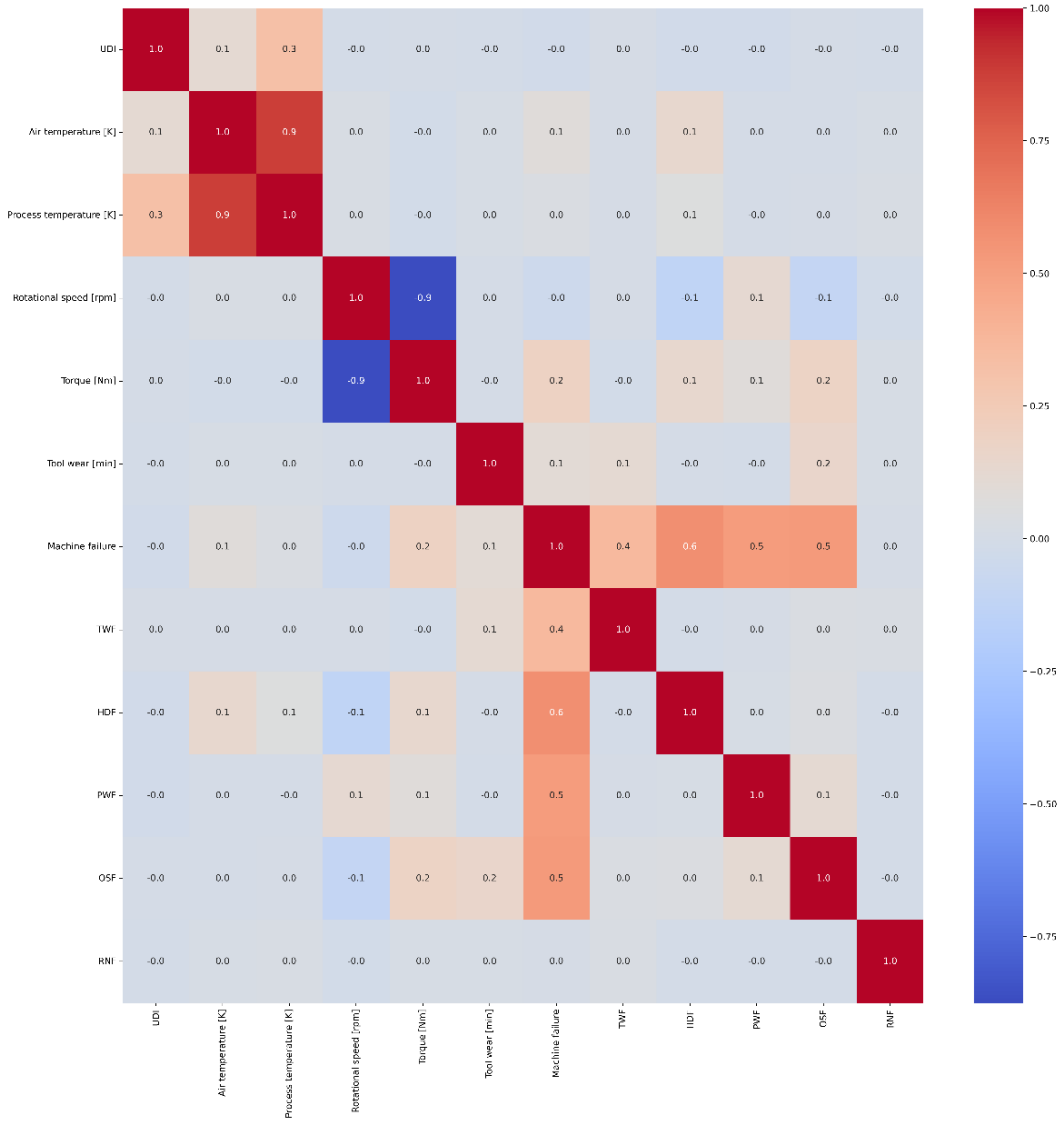
```
In [5]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 14 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   UDI                      10000 non-null  int64
 1   Product ID               10000 non-null  object
 2   Type                     10000 non-null  object
 3   Air temperature [K]      10000 non-null  float64
 4   Process temperature [K]  10000 non-null  float64
 5   Rotational speed [rpm]   10000 non-null  int64
 6   Torque [Nm]              10000 non-null  float64
 7   Tool wear [min]          10000 non-null  int64
 8   Machine failure          10000 non-null  int64
 9   TWF                      10000 non-null  int64
 10  HDF                      10000 non-null  int64
 11  PWF                      10000 non-null  int64
 12  OSF                      10000 non-null  int64
 13  RNF                      10000 non-null  int64
dtypes: float64(3), int64(9), object(2)
memory usage: 1.1+ MB
```

The dataset is highly imbalanced with number of machine failure(1) very less compared to the number of datapoints which do not have machine failure(0).

DATA VISUALIZATION

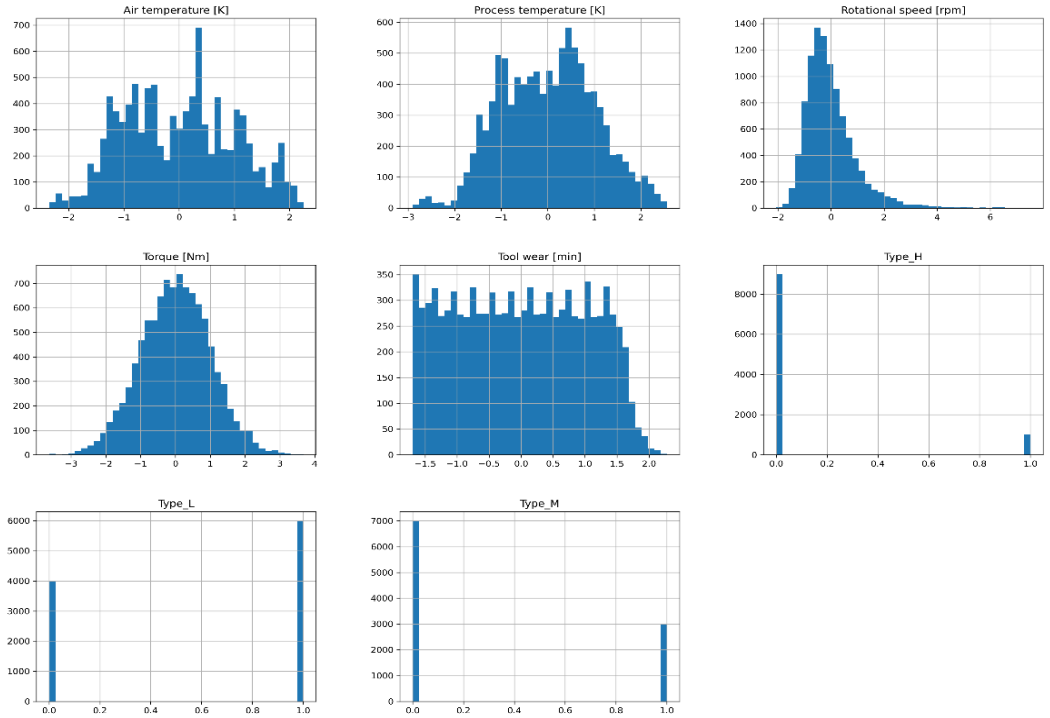
To visualize the data, histogram, heatmap of correlation and scatter plots are plotted.

Heatmap of correlation of all the input and output variables



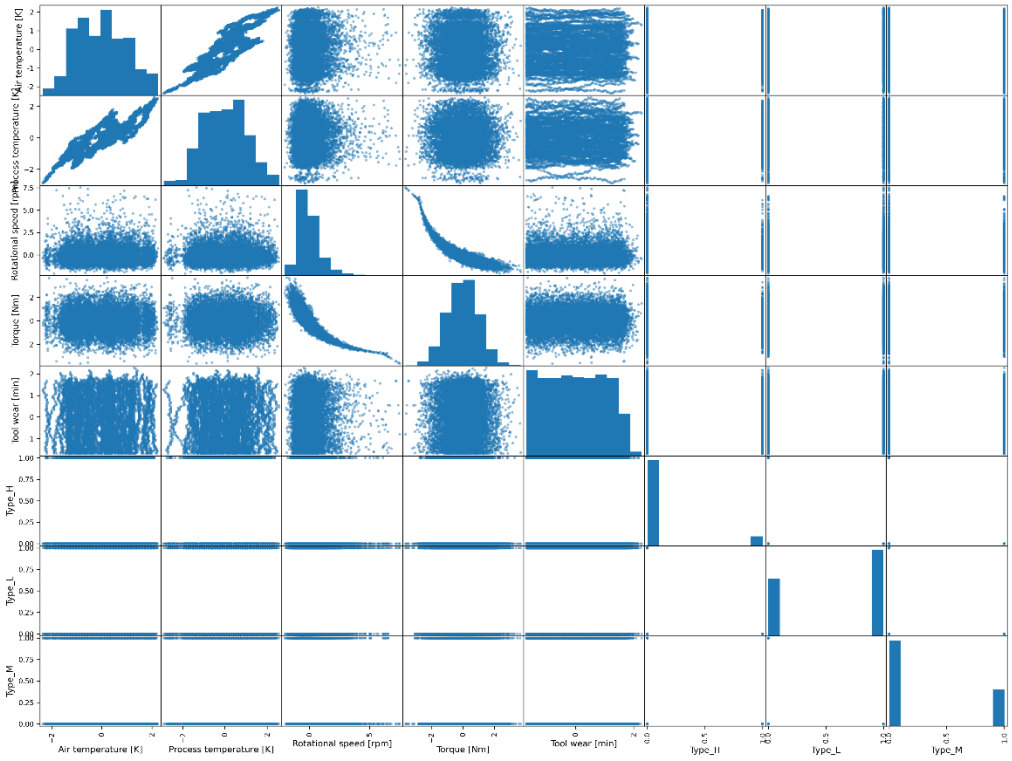
The above image shows the correlation between input variables and the input variables with the output variable.

Histogram plot of input variables



The above image shows the histogram plot of the input variables without any pre-processing. Some variables are omitted since they are categorical varaibles and they have to be quantified to be plotted on the graph.

Scatter plot of input and output variables



The above image shows the scatter plots among the input variables

From the correlation matrix and scatter plots, it is observed that 'Air temperature' and 'Process Temperature' are highly correlated with a correlation co-efficient of 0.9.

'Torque' and 'Rotational Speed' are inversely correlated with a correlation co-efficient of 0.9.

'Rotational Speed' is skewed to the right. There are no outliers in the dataset.

DATA CLEANING AND PRE-PROCESSING

The ‘Machine Failure’ column is dropped as another output variable ‘MF’ is created which performs ‘OR’ operation of the five independent failure modes(‘TWF’, ‘HDF’, ‘PWF’, ‘OSF’ and ‘RNF’).

The five independent failure modes i.e., ‘TWF’, ‘HDF’, ‘PWF’, ‘OSF’ and ‘RNF’ are dropped since they are already used to calculate ‘MF’ which is the output variable. They do not contribute to the prediction in any way.

The ‘UDI’ column is dropped since it just contains the index/numbering of the datapoints.

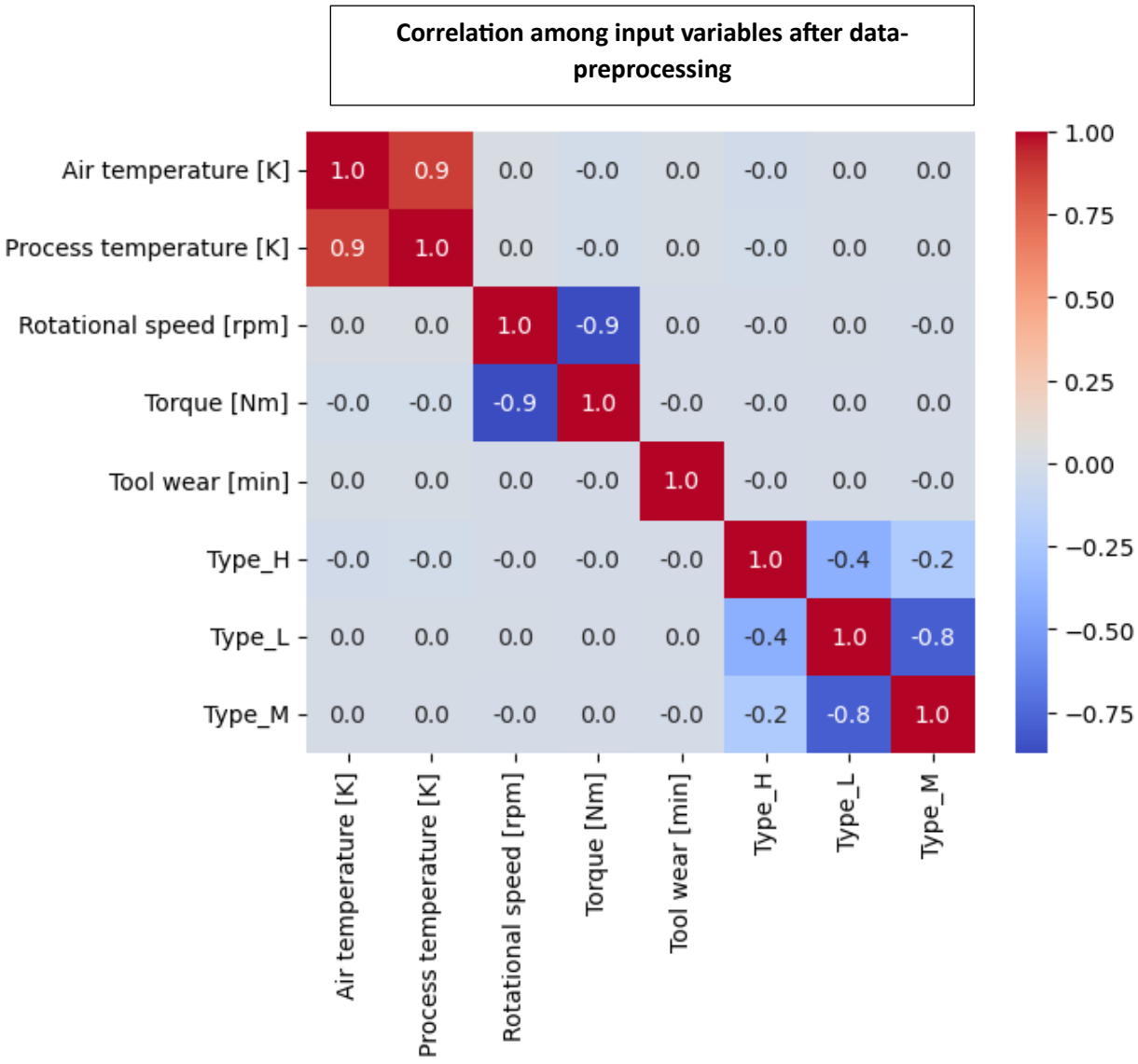
The ‘Product ID’ column is also dropped since it just contains a label/identifier of different datapoints.

The ‘Type’ column contains categorical variables and all the categorical variables have equal importance. Therefore, it is one-hot coded to Type_L, Type_M and Type_H.

The description of the dataset after pre-processing:

```
In [26]: inputData.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 8 columns):
 #   Column                   Non-Null Count  Dtype  
---  ------                   --------------  -----  
 0   Air temperature [K]      10000 non-null  float64
 1   Process temperature [K]  10000 non-null  float64
 2   Rotational speed [rpm]   10000 non-null  float64
 3   Torque [Nm]              10000 non-null  float64
 4   Tool wear [min]          10000 non-null  float64
 5   Type_H                   10000 non-null  float64
 6   Type_L                   10000 non-null  float64
 7   Type_M                   10000 non-null  float64
dtypes: float64(8)
memory usage: 625.1 KB
```

After data cleaning and data-preprocessing, there are 8 input variables and 1 output variable.



The above image shows the correlation between the input variables.

RELATED WORK

The paper titled “Explainable Artificial Intelligence for Predictive Maintenance Applications” is written by Stephan Matzka and generated by the authors of this paper for the purpose of developing a explainable machine learning model that will provide transparent and interpretable explanations for the predictions. The author aims to develop interpretable models using this dataset, unlike many machine learning models that are difficult to interpret since they are like black-boxes.

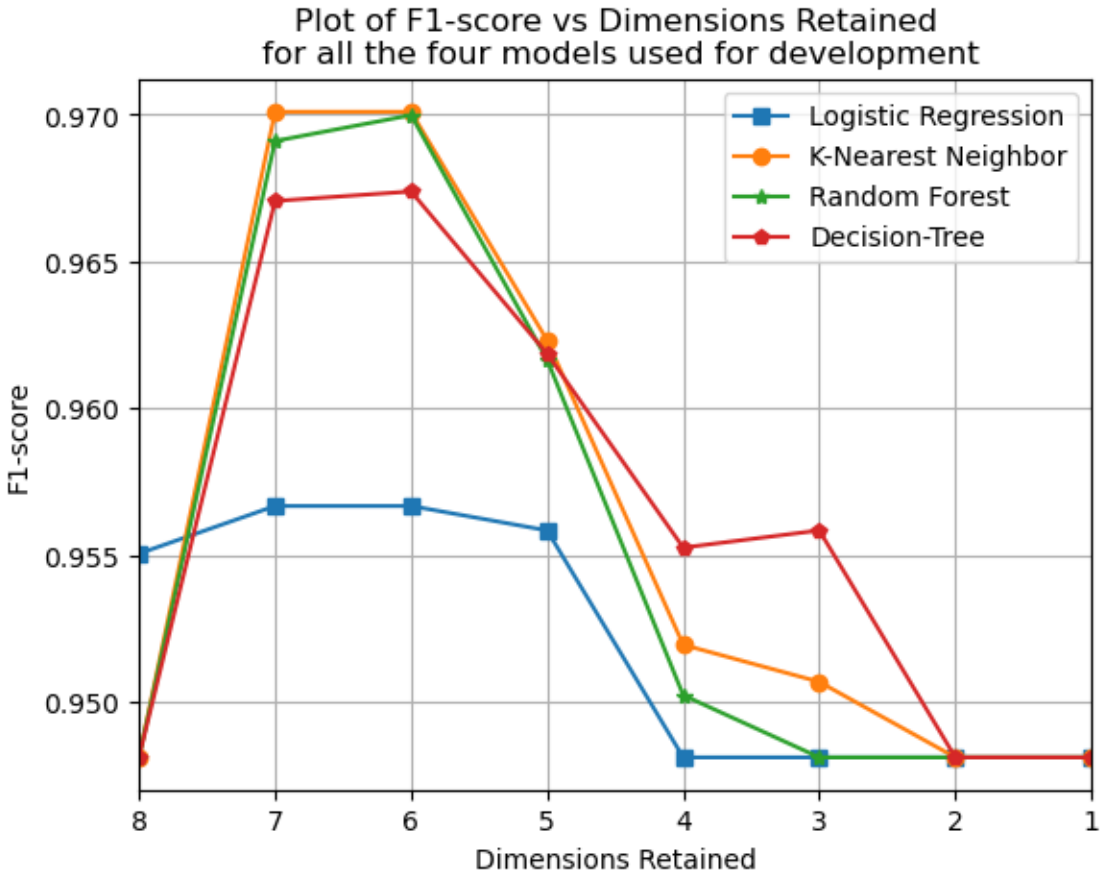
Two models are trained in this paper. First one is the bagged tree ensemble classifier using 5 fold cross validation. Relative predictor importance of the features used in the complex decision tree is obtained to find the important features. The second model a set of 15 decision trees that are trained with not all, but some features to obtain a maximum of 4 nodes to make interpretations for humans easier.

Two different approaches are used to explain the results. First one is the set of explainable decision trees and the second one is the normalized feature deviation.

To summarize, two methods are used provide an explanation for the classification result of a complex ensemble classifier and are evaluated on the synthetic predictive maintenance dataset. The decision trees provide better explanation compared to the normalized feature deviation but do not provide any explanation in some of the cases. Normalized feature deviations give sufficient explanation which is consisten although of lower quality.

FEATURE EXTRACTION

Feature selection is performed using backward search analysis. There are 8 input features after pre-processing. Backward selection begins with 8 input features, one feature is eliminated at each step and continues until no features are left. At each step, the F1-score is calculated and a graph of f1-score versus number of features is plotted. The features which reduce the F1-score are removed and combination of features with highest F1-score are retained. The graph of 'F1-score' versus 'number of features retained' is plotted.



In the above figure, it is observed that the F1-score is maximum when 6 inputs are retained for all the four models i.e., logistic regression, decision tree, random forest and k-Nearest neighbor. The plot yields the highest F1-score when the same input features are eliminated for all the four models. 'Air temperature' and 'Type_H' are the two input features eliminated for all the models.

MODEL DEVELOPMENT

Four different models are developed to predict machine failure. Logistic Regression, Random Forest, Decision Tree and k-Nearest Neighbor are the models used for development in this work to provide a exhaustive analysis of the prediction.

The entire data is split into two parts: Training set and Test set in the ratio 0.75 or 3 :1. Of the 10,000 datapoints, 7500 datapoints are used for training and 2500 datapoints are used for testing the performance of the model. The data is standardized for consistency. The dataset is split into training and test set in such a way that proportional number of datapoints of each type i.e., machine failure (1) and no machine failure(0) are distributed to the training and test sets. A random seed is set so that the same data is picked every time the model is run and to obtain consistent results. F1-score is selected as the evaluation metric since the dataset is highly imbalanced.

I. Logistic Regression: Logistic regression model is a simple model that is used for a binary classification problem. Since the problem statement is a binary classification problem, logistic regression is chosen as one of the models for development. The in-built sklearn library is used for this purpose. The machine failure is predicted based on a threshold value. The logistic regression model is fit for the training set and the f1-score is evaluated.

II. Random Forest: Random Forest is the second model used for performing predictions on the dataset. It is an ensemble classification model where an ensemble of trees is produced and combined results are obtained as predictions. Cross validation is performed to optimize the model.

III. Decision Tree: Decision Tree is a tree-based approach to performing predictions for classification problem. Decision Trees work by recursively splitting the feature space into decision regions.

IV. k-Nearest Neighbor: k-Nearest neighbor is a non-parametric model used to develop predictions for this dataset. There are no parameters that are optimized in this mode. Predictions are made based on the number of neighbors considered.

FINE-TUNING THE MODELS AND FEATURE SET

Fine tuning the models and the feature set includes experimenting with several values for different hyperparameters for a particular model being developed. Fine tuning for the different models is as follows:

Logistic Regression: The logistic regression model is optimized by using nested cross-validation wherein the test set is separated as a holdout set. An outer loop is used to optimize the model performance(F1-score) and an inner loop is used to optimize the hyperparameters. The hyperparameters optimized are regularization strength(C) and the type of regularization used(l1 and l2). Finally, the entire training set is trained using the optimized model and hyperparameters. The model is tested using the test set.

Random Forest: The Random Forest model is fine-tuned using grid search for four hyperparameters i.e., number of estimators, maximum depth, minimum samples splits and minimum number samples. The best parameters are found and used to fit the model using the training set.

Decision Tree: The decision tree is fine-tuned using grid-search for four three hyperparameters i.e., maximum depth, minimum number of samples and minimum number samples leaf. The best hyperparameters are found and fit to the model on the training set.

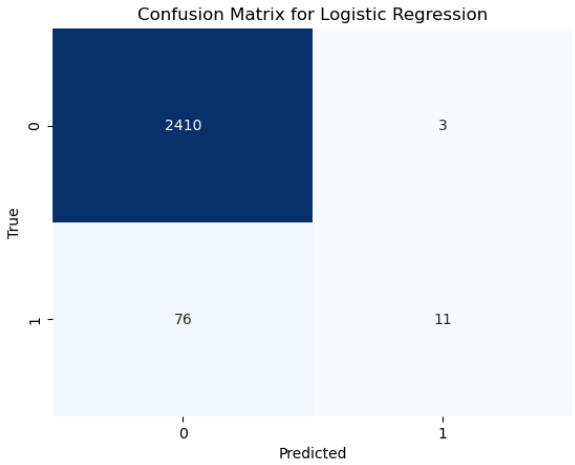
K-Nearest Neighbor: The K-Nearest Neighbor is fine-tuned for the number of neighbors parameter. The best parameters are found. The training set is re-trained using the training set

PERFORMANCE

All the four models are developed and fine-tuned in the previous steps and tested on the test set to obtain the F1-score for the test set. Various performance evaluation metrics such as accuracy, precision, recall and F1-score are used for classification problems to test their performance. F1-score is used to optimize the models since the dataset is highly imbalanced.

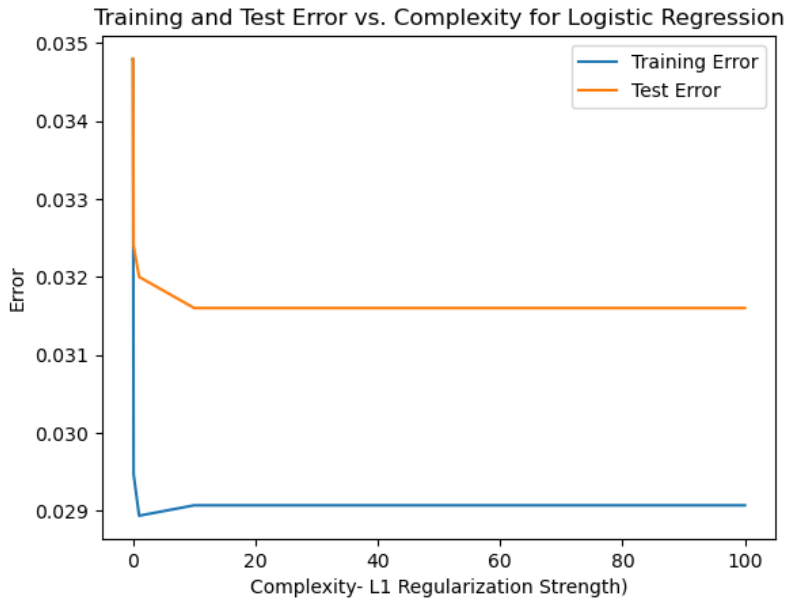
1. Logistic Regression:

Evaluation Metric	**Output**
Accuracy	0.97
Precision	0.96
Recall	0.97
F1-score	0.96



The confusion matrix for the logistic regression is plotted above. It is very clear that the dataset is highly imbalanced, and the number of false positives, false negatives, true negatives and false negatives is shown. The number of false predictions are less.

It is observed that the logistic regression model gives a F1-score of 0.96 on the validation set as well as the training set. Although regularization is applied, the model performance does not improve.

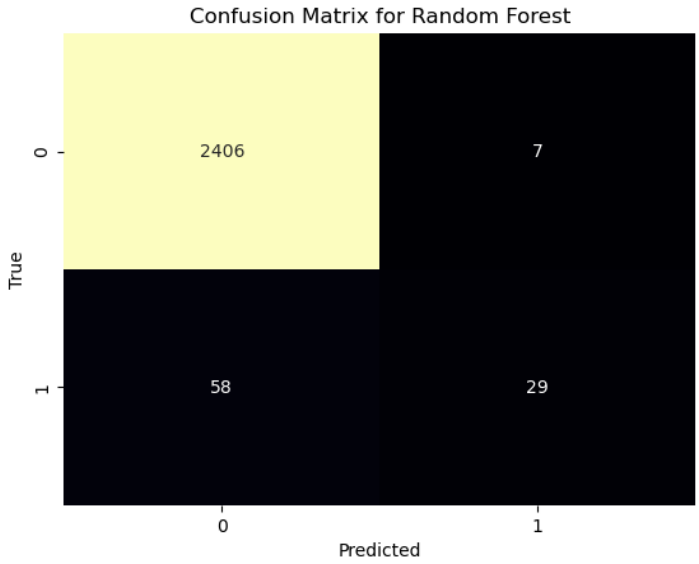


The above graph shows the train and test error as the regularization strength increases. It saturates beyond a certain point.
Best parameters for logistic regression:

Best parameters: {'C': 10.0, 'penalty': 'l1', 'solver': 'liblinear'}

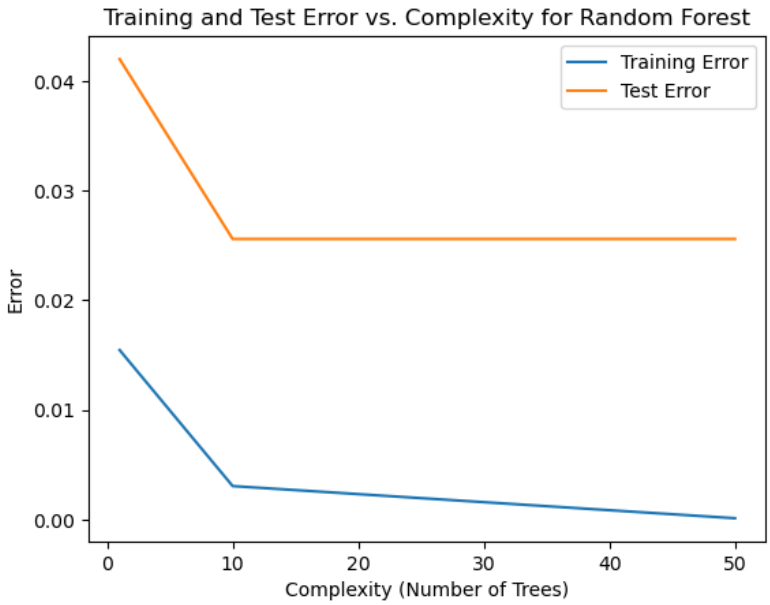
2. Random Forest:

Evaluation Metric	**Output**
Accuracy	0.97
Precision	0.97
Recall	0.97
F1-score	0.97



The confusion matrix for Random Forest is plotted above. It is very clear that the dataset is highly imbalanced, and the number of false positives, false negatives, true negatives and false negatives is shown. The number of false predictions are less.

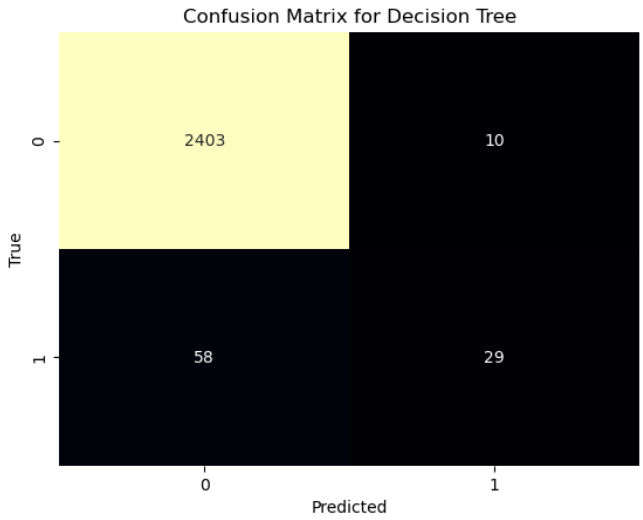
It is observed that for Random Forest model gives a F1-score of 0.97 on the validation set as well as the training set. Although regularization is applied, the model performance does not improve.



The above graph shows the train and test error as the complexity increases. It saturates beyond a certain point.

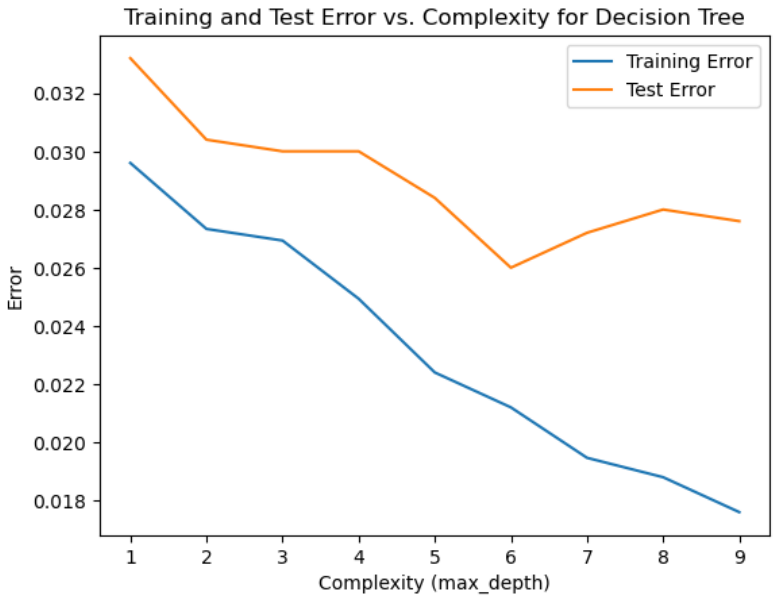
3. Decision Tree:

Evaluation Metric	**Output**
Accuracy	0.97
Precision	0.97
Recall	0.97
F1-score	0.97



The confusion matrix for Decision tree is plotted above. It is very clear that the dataset is highly imbalanced, and the number of false positives, false negatives, true negatives, and false negatives is shown. The number of false predictions is less.

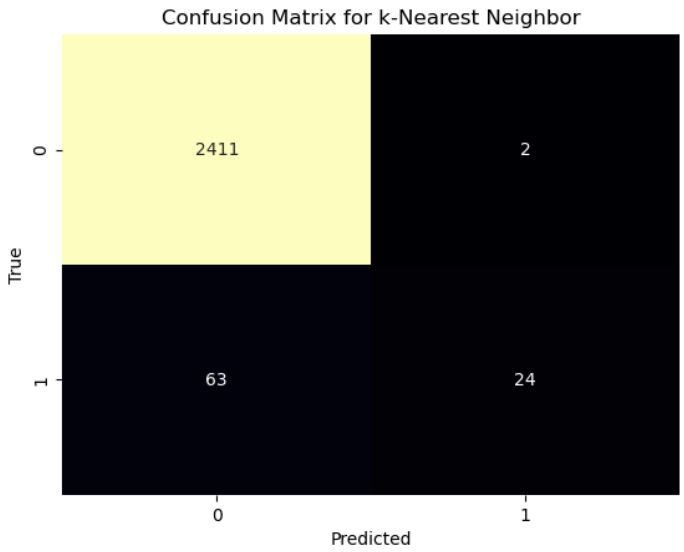
It is observed that the Decision tree model has a accuracy, precision, recall and a F1-score of 0.97 on the validation set as well as the training set. Although regularization is applied, the model performance does not improve much.



The above graph shows the train and test error as the complexity increases for decision tree. It saturates beyond a certain point.

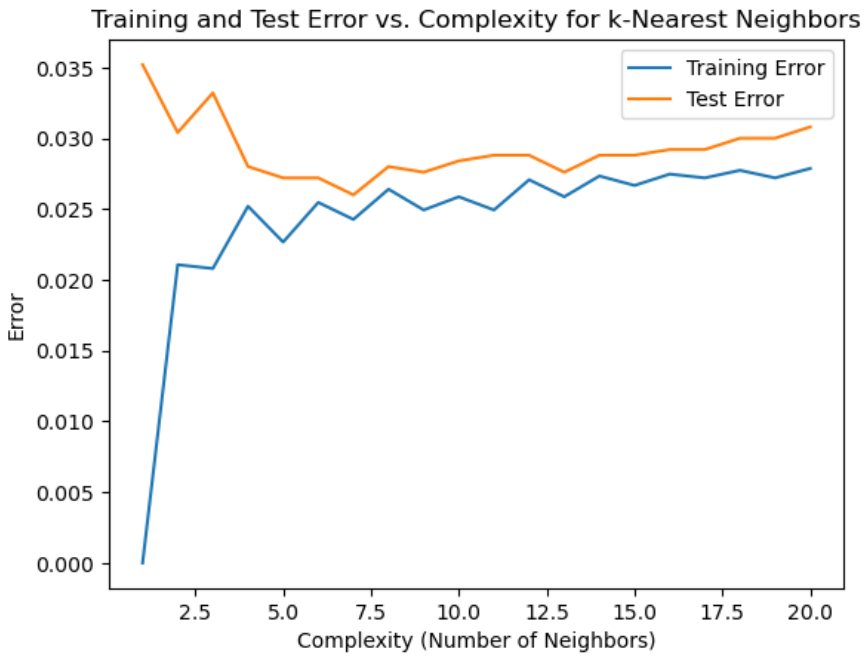
4. K-Nearest Neighbor:

Evaluation Metric	**Output**
Accuracy	0.97
Precision	0.97
Recall	0.97
F1-score	0.97



The confusion matrix for K-Nearest Neighbor is plotted above. It is very clear that the dataset is highly imbalanced, and the number of false positives, false negatives, true negatives, and false negatives is shown. The number of false predictions is less.

It is observed that the K-Nearest Neighbor model has a accuracy, precision, recall and a F1-score of 0.97 on the validation set as well as the training set. Although regularization is applied, the model performance does not improve much.



The above graph shows the train and test error as the complexity increases for K-Nearest Neighbor. It saturates beyond a certain point.

RESULTS AND CONCLUSION

The AI4I predictive maintenance dataset is fit for four different models. Feature selection is performed using backward search analysis. All the models provide very similar results when optimized for hyperparameters and have almost the same error for the validation set and perform very similarly on the test set. Applying regularization does not help much to improve the model. This may be due to the nature of the dataset as described in the related work.