University of Sheffield

Machine Learning for Advanced Manufacturing Applications

Naga Likith Surapaneni

Supervisor: John Clark

Module Code: COMU39/C

A report submitted in partial fulfilment of the requirements

for the degree of BSC AI and Computer Science

in the

Department of Computer Science

Declaration

All sentences or passages quoted in this report from other people's work have been specifically acknowledged by clear cross-referencing to author, work and page(s). Any illustrations that are not the work of the author of this report have been used with the explicit permission of the originator and are specifically acknowledged. I understand that failure to do this amounts to plagiarism and will be considered grounds for failure in this project and the degree examination as a whole.

Name: Naga Likith Surapaneni

COVID-19 Impact Statement

The lockdown imposed because of COVID-19 caused additional challenges for the completion of this project. In the second semester of the project, the university switched to online delivery of all teaching, and university buildings were closed. All project meetings were shifted to email correspondence and video meetings.

Real-time monitoring of the system was not possible to check if the tool monitoring was working accurately.

Table of Contents

Table of Figures

Chapter 1: Introduction

1.1 Overview
In this report, we are going to discuss machine learning in the manufacturing industry as it is rapidly developing in terms of producing readily available data and is going through an industrial revolution known as industry 4.0. Industry 4.0 is known differently across different countries for example in Germany it is called industry 4.0 and in Japan is already popularizing the idea of "Society 5.0" which tackles complex problems using industry 4.0.

This report also covers different machine learning techniques which are divided into two categories: supervised learning and unsupervised learning. The report also briefly talks about data mining. It is possible to make complex software systems that can manage important assets of a manufacturing industry plant. We can manage unplanned downtime and regularly check up on the health of different tools in the plant, this maintenance is known as predictive maintenance which is discussed further in the below chapters. The focus of this report is going to be on tool wear classification during the turning of A-1061 alloy by high-speed cutting tools. Using the provided dataset different machine learning algorithms and techniques can be compared and analysed to find out which machine learning technique is best suited for this dataset. This report aims to make a classifier that can accurately predict the tool wear and when it needs to be replaced to prevent unplanned downtime and help improve maintenance of the machining process.

1.2 Aim & Objective.
To summarize the objectives

- Research and get an idea of how to tackle the problem.
- Learn existing ML algorithms and libraries to use.
- Make an ML model that can predict tool condition.
- Do a comparative study with these ML techniques.

Chapter 2: Literature Review

2.1.1 Introduction to Industry 4.0

The world we know today has been shaped by 3 major technological revolutions. The first being the industrial revolution at the end of the 18th century. It was a transition to a new manufacturing process. This was followed by the second industrial revolution which led to the start-up of many factories in the early 20th century, and this led to an age of affordable consumer products for mass consumption. The third industrial revolution is the digital revolution which began in the 1950s which meant things that used to be analogue moved to digital technologies. There was the introduction of personal computers and included evolving online publishing platforms. Electronics and information technology began to take supply chains global. The digital era also opens the door to a new age of optimized and automated production.

In the twenty-first century, we have progressed to the fourth industrial revolution, which promises to combine the worlds of manufacturing and network communication in "The Internet of Things". Industry 4.0 reflects a technologically enabled paradigm change from centralised to decentralised output. Simply put, it implies that modern manufacturing machinery no longer simply processes the product, but that the product interacts with the machinery. Smart manufacturing processes and embedded manufacturing technologies are expected to usher in a new technological era that will change industry, production value chains, and business models, as well as build new smart factories. Figure 2.1 shows us a framework of the Industry 4.0 intelligent manufacturing system, in which research topics are categorized into smart design, smart machines, smart monitoring, smart control, and smart scheduling [1].

Figure 1

2.1.2 Application of Industry 4.0

Smart technology and Industry 4.0 application suggest new paradigms in industrial management and manufacturing. The German Platform 4.0 [2] addressed nine application scenarios to have a perspective on realistic implementations. Table 1 summarises and discusses the application cases.

Table 1 Application scenarios of Industry 4.0 [2]

Application scenario	Description
Adaptable Factory	This implementation situation is mostly concerned with the familiar principle of "Plug-and-Produce." This represents a factory's adaptability by physical transfer due to modularity, which is especially appealing for businesses whose orders are motivated by product originality and unpredictable demand.
Self-Organizing Adaptive Logistics	This scenario can be thought of as a logistics infrastructure that can operate without direct human interaction, needing more decentralised, agile, and independently interacting materials processing modules and efficient delivery logistics.
Transparency and Adaptability of Delivered Products	Delivered products gather data on their own to optimise market operations, developing new business models, and dynamically adapting product functionality. This implementation scenario explains the evolution of devices capable of adaptable design.
Smart Product Development for Smart Production	The most critical aspect of this application situation is the common product engineering. Virtual manufactured goods allow new modes of engineering cooperation as well as the automation of engineering tasks.
Innovative Product Development	Several players are now collaborating to develop innovative and updated products, as well as skills, service and specialist technology vendors, and product customers. This is enabled by emerging ways of Internet-based collaboration.
Operator Support in Production	In addition to all other potential fields of employment, humans are aided by the increasing digitisation of industrial manufacturing. This boost both enthusiasm and productivity. Humans may analyse more difficult scenarios more quickly and

	thoroughly, paying closer attention to the main considerations and details of their tasks.
Order-Controlled Production	In this scenario, the organisation could easily adjust to its portfolio, particularly it's manufacturing, to enlarge innovation and shorten product cycles, and make the best use of existing production facilities' resources and capabilities ad hoc according to demand.

Germany has long been the world's leading innovator in the automotive sector, and it is only in Germany that the climate is ideal for this modern change: Industry 4.0 is being introduced: The technology that, soon, has the potential to change the manufacturing population all over the world.

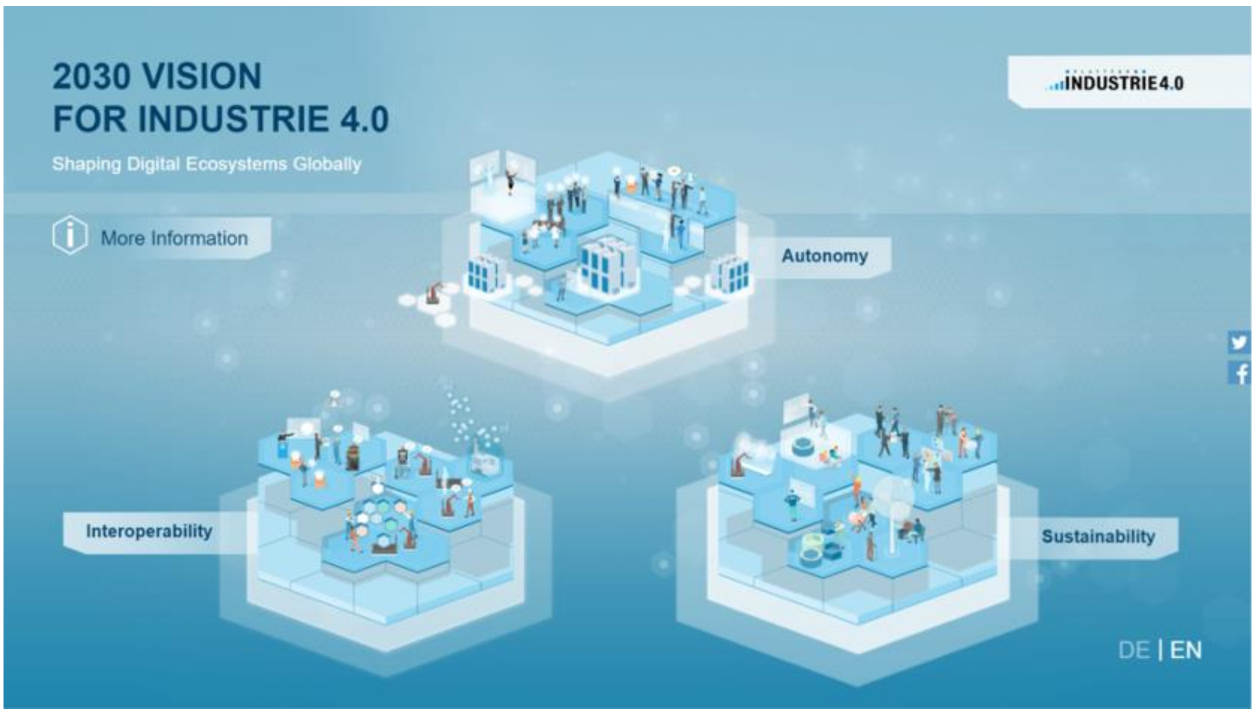

Figure 2

2.1.3 Impact of Industry 4.0 on Manufacturing Industry

Productivity and efficiency

The growing number of smart devices in the factory, as well as the efficiency with which they communicate with one another, are favouring global plant streamlining, resulting in an improvement in industrial productivity. Indeed, a collaboration between humans, between people and smart devices (Human-computer Interaction, HCI), and between smart devices themselves (Machine to machine, M2M) enables collective efficiency [3]. The entire development process can be streamlined with the help of big data and M2M; typical manufacturing routes are leaner, and resource consumption level is increased. Productivity is increased due to the versatility provided by smart machines' self-organization function, which allows them to automatically reconfigure themselves to generate various types of items.

Big Data analytics can provide a precise understanding of the manufacturing process and measure success metrics for devices, goods, and processes. With this detail, the worker can effectively schedule production and make further precise judgments in a shorter period.

Supply chain management

The supply chain would be fully automated, supported by interconnected networks, and perfectly organised because of the amalgamation of internet technology with information flow and self-monitoring smart machines and smart products. Both factors would result in cost savings, increased process transparency, procurement process management, and stability, especially in the purchasing, manufacturing, and delivery stages. Smart devices in “Procurement 4.0” [4] will assess the need for a certain commodity and produce an order that is autonomously delivered to the merchant without the need for human interference. Furthermore, distribution operations are distinguished by a high degree of automation, implying autonomous judgment-making, monitoring, and logistics organizing.

Quality Management

Because of the core facets of vertical integration, horizontal integration, and end-to-end engineering development, Industry 4.0 provides significant opportunities in quality management [5]. Vertical integration connects all the methods in the various manufacturing stages, providing invaluable information for monitoring performance and regulating processes in real-time. In the event of an output variance triggered by a system failure, sensors automatically alert the responsible party so that the problem can be quickly identified and repaired. Horizontal convergence enables the user to obtain the product's existing state at any time. Finally, engineers can analyse product behaviour at any moment and adjust easily.

2.2 Tool Condition Monitoring

Tool wear in the metal cutting process refers to the progressive failure of cutting tools because of routine activity. As the procedure continues, the amount of tool wears increases, reducing tool life. Estimating the remaining useful life of industrial products is a crucial activity in the manufacturing process. Early identification of tool deterioration allows for the reduction of unintended loss, reducing production costs and increasing efficiency. It also aids in the preservation of the workpiece's accuracy. It has been shown that there is a relationship between workpiece roughness and cutting tool wear. Given this, comprehensive research is being performed worldwide, and many approaches for tool conditioning monitoring systems have been suggested by researchers.

These are classified into 2 main groups: Direct methods and indirect methods [6]. The direct tool wear methods have the advantages of capturing actual geometric changes from the wear of the tool. However, direct measurements are less beneficial because the cutting area is largely inaccessible and continuously in contact between the tool and workpiece. Indirect methods are based on parameters measured during the cutting operation which can be related to the wear state. In the indirect method, the condition of the cutting tool is predicted by processing the output sensor signals.

Here Table 2 shows the direct and indirect tool wear sensing methods.

Table 2

Direct Method	Indirect Method
Electric Resistance	Cutting Force
Optical	Vibration
Radioactive	Torque/current etc.

In general, when using the indirect method for Tool Condition Monitoring it mainly consists of four steps. [6]

- Data collection in the form of sensor signals Cutting power, friction, temperature, and/or motor current are some examples.
- Signal analysis and extraction of valuable features
- Instrument wear classification or prediction using pattern analysis, machine learning algorithms, or neural networks.
- Development of a decision-making technique to monitor the machining process using sensors.

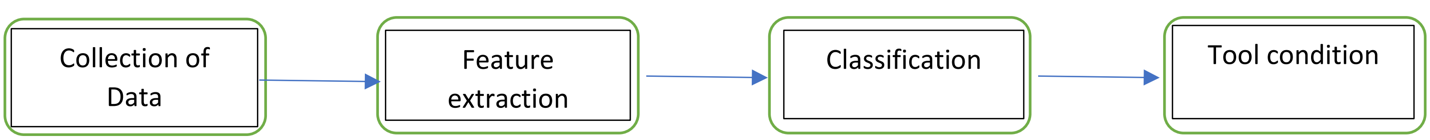

Figure 3

Monitoring and controlling industrial processes are increasingly becoming a driving force in the growth and survival of manufacturing sectors. AC (Adaptive Control) systems have the potential to improve the controllability and efficiency of machining processes. AC has been in use since the 1960s, but it has not yet evolved enough to dominate the manufacturing world. Because of the environmental effects, legislation, and regulations, the need for consistency, production, and sustainability in manufacturing is more critical than ever before. As a result, the need for incorporating AC into production lines, for online optimization, not just in terms of tool conditioning, but also in terms of making manufacturing more competitive and sustainable, is critical.

The production and deployment of advanced AC solutions will be critical in the future improvement of machining systems and their operating efficiency. To satisfy the demands of advanced industrial technology, these novel structures must be durable, reconfigurable, efficient, insightful, and inexpensive.

2.3 Predictive maintenance

Maintenance costs are a major part of operating costs in the manufacturing industry. Maintenance costs can go up to 60% of the cost of the goods produced [7]. Recent surveys of maintenance management effectiveness show that 1/3 of all maintenance costs is wasted. US industry spends more than 200 billion dollars each year on maintenance of plant equipment. The ineffectiveness of the management poses a loss of more than 60 billion dollars each year, showing the importance of maintenance.

Predictive maintenance is a technique for increasing the performance, quality, and overall efficiency of manufacturing and processing factories. A reliable predictive maintenance programme determines a system's operational status by using the most cost-effective procedures (e.g., vibration monitoring). In predictive maintenance, the direct influence of mechanical condition, system activity, and other functionality is often used to measure the loss efficiency or mean time to failure for each system in the plant.

The maintenance schedule of computers can adjust as they age and based on their level of use, which can be handled by predictive maintenance. Over time, various parts of the system will react differently to production stress. The subsequent rise in maintenance expected by data trends will show whether a system has reached a cost-performance tipping point. The need to potentially repair major sections of a system, or the whole device, is made manageable by the ability to predict and prepare for the need, both in terms of expense/expenditure and time/effort.

A successful maintenance programme will reduce unscheduled breakdowns in all technical equipment and ensure that fixed equipment remains in good working order. Most technical issues can be avoided if they are identified early.

Companies like GE (General Electric) have a mature platform where AI is used for Predictive maintenance in the industry. GE Digital has a product called Predix asset performance management which is a suite of software and service solutions that aims to help optimize the performance of assets. It uses advanced analytics to turn data into actionable insights [8]. The APM has helped a fertilizer manufacturing company in Brazil called Vale Fertilizantes save $1.4M and avoided 25 days of lost production.

2.4 Machine Learning in the manufacturing industry.

The automotive sector has been generating a lot of new data. Data comes in a variety of formats and semantics, such as sensor data from a manufacturing line, environmental data, and machine tool parameters. This rise in the abundance of data is often referred to as Big Data. The advancement of mathematical and computer science instruments, which are often made publicly available, can completely change the industry. Machine learning is one of the most promising inventions (ML).

ML strategies can aid in automating the time-consuming process of information acquisition, which is required to construct a knowledge base structure. Automation can speed up and lower the cost of construction by reducing the amount of time (and sometimes labour) required. [9]. ML has been widely applied in a wide range of industrial applications, from tracking and management to process optimization and predictive maintenance. ML techniques have been discovered to have promising potential in manufacturing processes for improved quality control optimization, especially in a complex manufacturing setting were determining the root cause of problems is difficult. [10]. ML has mostly been used to target discrete issues rather than the whole production chain. It is a guide that allows one to have a better understanding of the domain. Another benefit of ML algorithm is their ability to deal with high-dimensional problems and results [10]. ML algorithms can identify previously unknown relationships in data sets.

Each problem has a unique set of requirements and choosing an appropriate ML algorithm is unique. The output of each algorithm is therefore dependent on the data available and the data pre-processing. The easiest way to choose an appropriate algorithm is to try to evaluate various algorithms. To choose the best-suited algorithm, they should be checked in practical settings.

2.5 Machine Learning

Machine learning is a form of artificial intelligence application (AI). It is characterised as computational methods that make use of experience to improve performance and make accurate predictions. In this context, experience refers to a type of electronic evidence that has been compiled and made available for study. The primary goal of machine learning is to make reliable predictions by constructing effective and scalable algorithms.

Approaches

2.5.1 Supervised Learning & Unsupervised Learning

Supervised learning provides you with tools to classify data. In any data set used by machine learning algorithms, an instance is represented by using the same set of features. These features may be continuous, categorical, or binary. If these instances are given with known labels, then it is called supervised learning [11]. The task of a supervised learning algorithm aims to map an input with an output, which allows the model to learn over time. The accuracy of a supervised learning algorithm is measured through the loss function. In supervised learning, we aim to minimize the error for each training example during the learning process. A loss function maps options with the costs associated.

Supervised learning can be separated into two categories [12]:

- Classification uses an algorithm to assign test data into specific categories. It learns from the data set and tries to conclude how the entities should be labelled. Examples of classification algorithms include support vector machine (SVM), K-nearest neighbours and random forest.
- Regression draws out the relationship between independent and dependent variables. A few examples include linear regression, logistic regression, and polynomial regression.

Unsupervised learning is a type of machine learning where the data set does not have pre-existing labels; it tries to search for patterns and relationships with almost no human supervision. Unsupervised learning is more complex and more unpredictable compared to other learning methods. An example of unsupervised learning is clustering.

Clustering algorithms use biases that help in specific areas of possible groupings and relationships among a given data set. Actual problem-solving applications have focused on classification and are very flexible which means that they predict missing feature values in certain cases with very high accuracy [13].

2.6 Machine Learning Processes and Techniques

2.6.1 Support Vector Machine

A Support Vector Machine (SVM) is a supervised learning technique that creates boundaries separating different groups from a given data set. The algorithm creates a boundary, known as the hyperplane, that maximizes the distance between the two different classes in the data set. SVM can handle very large feature spaces as it can map data to high dimensional space where it is easier to classify with linear surfaces. SVM based classifiers are said to have good generalization properties as compared to normal classifiers because it focuses on reducing structural misclassification risk [14]. Recently support vector machines have been introduced for solving pattern recognition problems. The SVM maps the data into higher dimension input space and constructs an optimal separating hyperplane.

Given the training data set of N data points $\{y_k, x_k\}_{k=1}^{N}$, where $x_k \in R^n$ is the kth input and $y_k \in R^n$ kth output, the support vector machine approach aims at constructing a classifier of the form:

$$y(x) = sign\left[\sum_{k=1}^{N} \propto_k y_k \psi(x, x_k) + b\right],$$

Where $\propto_k$ are positive real constants and b is a real constant. For $\psi(.,.)$ typically has the following choices:

$\psi(x, x_k) = x_k^T x$ (linear SVM, this is an example where the hyperplane is linear); $\psi(x, x_k) = (x_k^T x + 1)^d$ (polynomial SVM of degree d); $\psi(x, x_k) = \exp\left\{-\frac{\|x - x_k\|_2^2}{\sigma^2}\right\}$ (Radial basis function/RBF SVM), where σ is a constant.

The classifier is constructed as follows.

$$w^{\mathrm{T}}\psi(x_k) + b \geq 1 \quad , \quad \text{if } y_k = +1,$$

$$w^{\mathrm{T}}\psi(x_k) + b \leq -1 \; , \quad \text{if } y_k = -1.$$

RBF kernels are the most generalised type of kernelization and, due to their similarities to the Gaussian distribution, are one of the most used kernels. The RBF kernel function computes the resemblance or how close two points X1 and X2 are to each other.

RBF Kernel is well-known due to its resemblance to the K-Nearest Neighbours Algorithm. It has the benefits of K-NN when overcoming the space complexity issue because RBF Kernel Support Vector Machines only need to store the support vectors during training rather than the whole dataset.

2.6.2 K – Means.

K means clustering algorithm is an unsupervised learning technique. K means the algorithm works by making centroids for different classes which are used to define clusters, if a data point is close to a centroid, then it belongs to that cluster. K means the algorithm changes the model to predict labels given features even when the data set does not have labels [15]. The centroid of a class may be fixed or not depending on the data set and how the K means algorithm has been programmed. The purpose of a centroid is to make sure that it is cleverly placed in such a way that it represents all the data points in the given class.

The distance between clusters is the distance between the centroids and the objective of K-Means clustering is to minimize total intra-cluster variance. Intra-cluster distance is the distance between two different objects belonging to two different clusters. Usually, this distance is larger than the inter-cluster distance which the distance between objects belonging to the same cluster. K-means is an effective method however we need to specify the number of clusters and the results may vary depending on the initialization and often terminates at a local optimum. In general, a large k might reduce the error but increases the risk of overfitting the data.

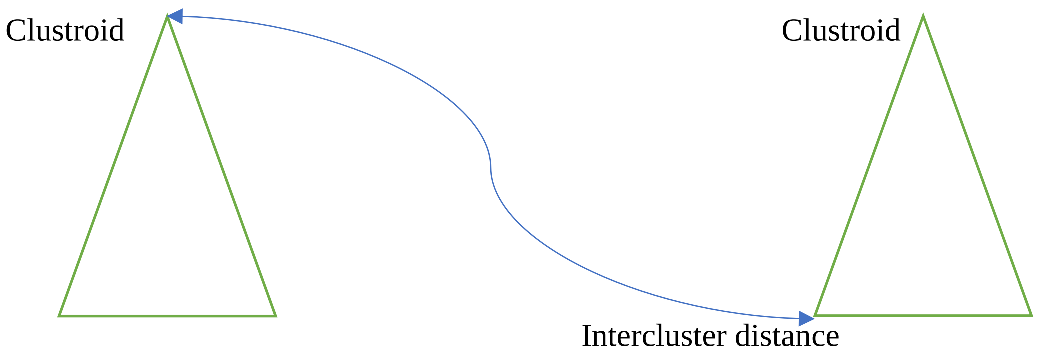

Figure 4

This algorithm aims to minimize an objective function, In this case, the squared error function [16].

$$W(S,C) = \sum_{k=1}^{K} \sum_{i \in S_k} \|y_i - c_k\|^2$$

Where S is a K-cluster partition represented by vectors y_i in the M-dimensional feature space, consisting of non-empty non-overlapping clusters S_k , each with a centroid c_k (k=1,2…. K).

2.6.3 Random Forest:

Random forest is an algorithm that can be used for a variety of tasks including regression and classification. It is an ensemble method which means that a random forest is made up of many decision trees. In a decision tree, there are two nodes which are the decision node and leaf node. Decision nodes are used to make any conclusions and have multiple branches, whereas a leaf node is the output of the decision nodes and do not contain any further branches. They are simple to understand and can be visualized. Usually, the decision trees are prone to overfitting to the training set and hence perform poorly in the test set. The random forest's ensemble design helps generalize well to unseen data and even data with missing values. The random forest algorithm searches for the best feature among the subset of features [17].

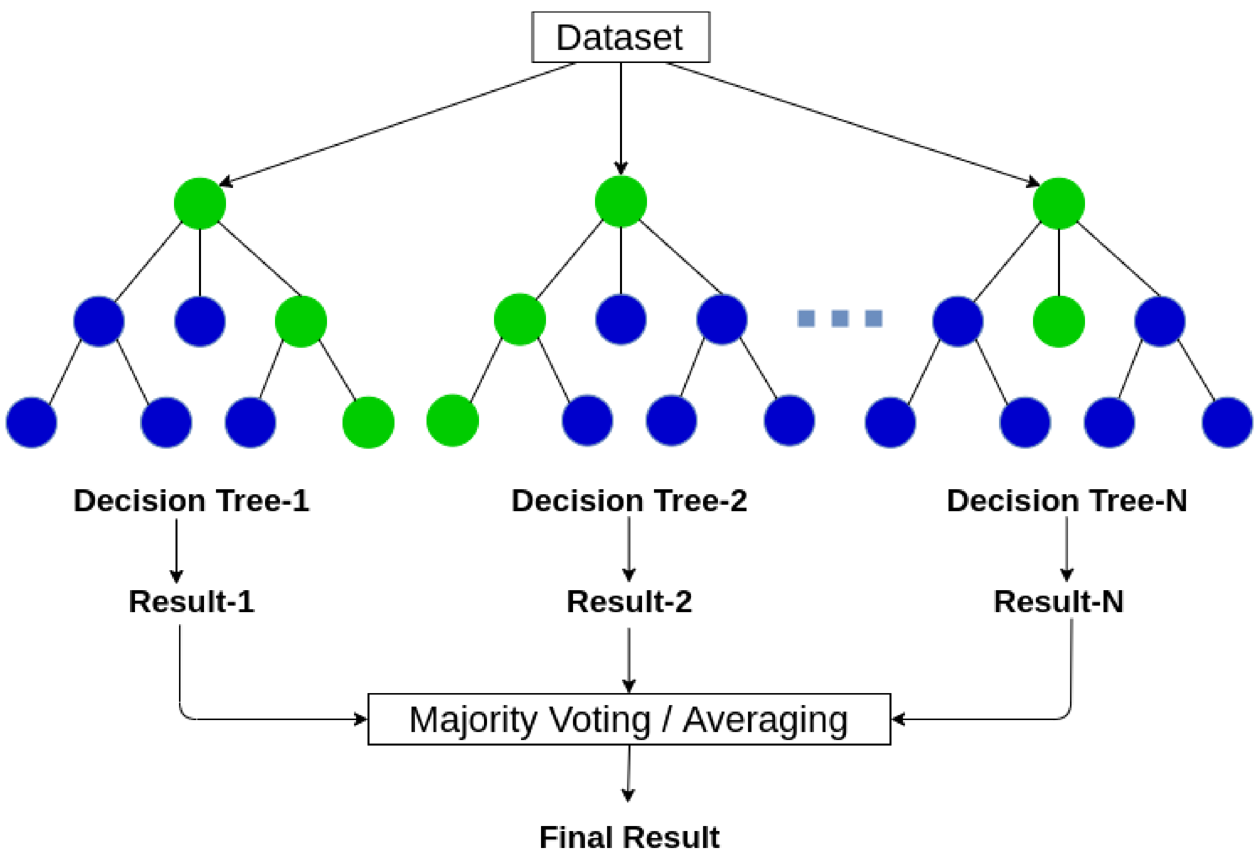

Figure 5

Important hyperparameters

These are the hyperparameters of sklearn's built-in random forest function. They help increase the predictive power of a model or makes the model run faster [18] .

Increasing the predictive power: "n_estimators", which is the number of trees in the forest the default being 100 [19]. The greater number of trees helps increase the predictive power but also makes the computational speed slow. Another important hyperparameter is "max_features", which is the number of features to consider when looking for the best split and the last important hyperparameter "min_sample_leaf", is the minimum number of samples required to be at a leaf node.

Increasing the model's speed: "n_jobs" hyperparameter is the number of jobs to run in parallel, it tells the engine how many processors it allowed to use. Lastly, there is "random_state", which controls the randomness of bootstrapping of the samples used when building trees. A model will produce the same result when there is a definite value of "random_state".

2.6.4 Ridge Regression

Ridge regression is a common parameter estimation method for dealing with the collinearity issue that frequently arises in multiple linear regression. Ridge regression is a subset of Tikhonov regularisation in which all parameters are similarly regularised. Collinearity has a well-known negative effect on the least-squares (LS) estimator in a regression sense. Many approaches to mitigating this impact were established, with many focusing on variable elimination. Ridge regression, on the other hand, addresses the issue of collinearity without excluding variables from the initial set of independent variables. In some applications, this proved to be a very appealing function. The OLS (Ordinary Least Squares) loss function is amplified in Ridge Regression in a way that it not only minimises the number of squared residuals but also penalise the scale of constraint estimates to shrink them to zero.

$$L_{ridge}(\beta) = \sum_{i-1}^{n}(y_i - x_i\beta)^2 + \lambda\sum_{j-1}^{m}\beta^2 = \|y - X\beta\|^2 + \lambda\|\beta\|^2$$

Where Y represents the dependent variable and X represents the independent variable. β is the regression coefficient.

Solving for β gives the ridge regression estimates, the λ parameter is the regularization penalty. Here is what happens when λ changes from zero to infinity.

$$\text{As } \lambda \to 0, \quad \beta_{ridge} \to \beta_{OLS}$$

$$\text{As } \lambda \rightarrow \infty, \quad \beta_{ridge} \rightarrow 0$$

2.6.5 Lasso Regression

For parameter estimation in regression problems, the Tibshirani lasso has become a common alternative to ordinary least squares. Its popularity can be attributed in part to a key function of the method: the shrinkage of the vector of regression coefficients toward zero, with the choice of setting those coefficients identically equal to zero, resulting in a simultaneous estimation and variable selection process. [20]. The lasso is a kind of penalised least-squares algorithm that minimises the residual sum of squares while managing the coefficient vector's (β) L1-norm.

$$\beta_L - argmin_\beta (y - X\beta)^T (y - X\beta) + \lambda \|\beta\|_1,$$

Where $\lambda \geq 0$ calculates the sum of shrinkage. The case $\lambda = 0$ results in $\beta_L = \beta_{OLS}$.

2.6.6 Gradient Boosting

Gradient boosting algorithms are a popular category of machine learning techniques that have shown significant performance in a wide variety of practical applications. They are highly adaptable to the specific needs of the application, such as learning concerning various loss functions. The learning technique in gradient boosting machines, or simply GBMs, adds new models in succession to provide a more reliable approximation of the response vector. The basic concept behind this algorithm is to create new fundamental learners that are maximally correlated with the negative gradient of the loss function, which is identical to the whole ensemble.

GBM’s can best be demonstrated by first discussing the Adaboost algorithm. The algorithm begins with training a decision tree of identical weights for every observation. After inspecting the first tree, it increases the weights of “difficult to classify” observations and reduces the weights of “easy to classify” observations. This results in the second tree are built on the weighted outcomes. The goal is to build on the predictions of the first tree. This method is repeated for several iterations. Successive trees help with the classification of outcomes that were not classified well by the preceding trees. The final ensemble model's predictions are baes on the weighted sum of the predictions made by the former tree models.

GBM trains multiple models gradually, additively, and in a sequence. The primary distinction between Adaboost and the GBM is the ability to deal with the shortcomings of slow learners (e.g., decision trees). While the Adaboost model identifies faults by utilizing high weight data points, GBM does by using gradients in the loss function*. The loss function is a metric that shows how well the model's coefficients correspond to the underlying results.

$y = ax + b + c$, where c is the error term*

In the function estimation problem, one has a method made up of a random "output" or "response" variable and a collection of random "data" or "explanatory" variables. $X = \{x_1, \dots, x_2\}$. Given a training sample $\{y_i, x_i\}_1^N$ of known (y, x) values, the goal is to find a function F*(x) that maps x to y such that the expected value of any defined loss function $\psi\ (y, F(x))$ is reduced over the combined distribution of all (y, x) values [21].

$$F^*(x) = \arg min_{F(x)}\ E_{y,x}\psi(y, F(x))$$

2.6.7 Xgboost

xgboost is an abbreviation for the eXtreme Gradient Boosting kit. It is a scalable and effective implementation of the gradient boosting paradigm by [22]. The bundle contains an effective linear model solver as well as a tree learning algorithm. It may perform a variety of objective functions, such as regression, sorting, and ranking. The package is designed to be extensible, allowing users to quickly identify their own goals.

It has several features:

1. Speed: xgboost can perform parallel computations automatically on Windows and Linux using OpenMP. It is normally more than ten times quicker than gradient boosting.
2. Sparsity: xgboost accepts sparse feedback and is designed for sparse input for both tree booster and linear booster.
3. Customization: xgboost allows you to build your target function and measurement function.
4. Performance: xgboost outperforms other algorithms on a variety of datasets.

2.6.8 Cat Boost (Categorical boosting)

Cat boost is a new gradient boosting algorithm that manages categorical features effectively and takes advantage of working with them during testing rather than pre-processing period. Another drawback of the algorithm is that when choosing the tree structure, it employs a new schema for measuring leaf values, which helps to minimise overfitting.

Cat Boost is available in CPU and GPU implementations. On ensembles of comparable scale, the GPU implementation makes for even quicker training and outperforms all state-of-the-art open-source GBDT GPU implementations, Xgboost and LightGBM. The library also includes a fast CPU scoring implementation that outperforms Xgboost and LightGBM implementations on similar-sized ensembles.

Chapter 3: Requirements and analysis

Introduction

From the report, we know that using machine learning in manufacturing industries in solving problems such as tool wear classification can save companies money and maintain the industrial plant equipment. This chapter aims to set out requirements for the project to define success criteria at the end of the project.

3.1 Functional Requirements

Implement software that to apply a wide range of ML techniques to the data set to predict the machine tool wear. There is scope for improvements in the maintenance systems and increasing the accuracy of the prediction model of tool wear. There is also scope for a comparative study between two ML approaches which are supervised and unsupervised learning in the context of the given dataset.

An investigation will be carried out to compare the techniques used to make software. This includes comparing the accuracy of supervised learning and non-supervised learning. The algorithms which will be used will include K-mean, Random Forest and SVM.

3.2 Model Evaluation Metris

3.2.1 Root Mean Squared Error

When you experiment that yields a set of observable values that you want to equate to theoretical values, the root-mean-square variance (RMSD) allows you to measure this relation. RMSD is calculated by taking the square root of the mean square error.

$$RMSD = \sqrt{\frac{\sum(x_e - x_o)^2}{n}}$$

for $x_e$predicted values, $x_o$observed values, and n cumulative number of values.

This way of calculating a variance (or deviation), squaring each difference, adding them all together, dividing by the number of data points (as you might when calculating the average of a series of data), and then taking the square root of the result is what gives the quantity its name, "root-mean-square deviation." [23] For tests, RMSD provides a specific, unified means of evaluating how errors of how expected values vary from observable values differ. The smaller the RMSD, the closer the experimental findings are to the theoretical projections. They allow you to measure the impact of different sources of error on the observed experimental effects.

3.2.2 Mean Squared Error

MSE, or Mean Squared Error, is a common error metric for regression problems.

It is also a significant loss function for algorithms that adapt or simplify regression problems using the least-squares framing. The term "least squares" refers to the goal of minimising the mean squared error between forecasts and predicted values.

The mean or average of the squared deviations between predicted and expected target values in a dataset is used to measure the MSE.

$$MSE = \frac{1}{N}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where Y_i is the dataset's i^{th} expected value and $\hat{Y}_i$ is the estimated value. The discrepancy between these two values is squared, which removes the sign and results in a positive error value.

Squaring has the additional benefit of inflating or magnifying significant errors. That is, the greater the difference between the predicted and expected values, the greater the squared positive error. When MSE is used as a loss function, this has the effect of "punishing" models further for greater errors. When used as a metric, it also has the purpose of "punishing" models by inflating the total error score.

3.2.3 R-Squared

Another parameter for measuring the efficiency of a regression model is the coefficient of determination or R2. The metric allows us to equate our current model to a constant benchmark to determine how much improved our model is [24]. The constant baseline is determined by taking the data's mean and drawing a line along with it. R2 is a scale-free score, which means that regardless of whether the values are too high or too small, the R2 will either be less than or equal to one.

$$R2 = 1 - \frac{MSE(model)}{MSE(baseline)}$$

A regression model is said to suit the data well if the deviations between the observed and expected values are small and unbiased. In this case, unbiased means that the fitted values are not uniformly too high or too low around the observation space.

R-squared is a linear regression process goodness-of-fit metric. This statistic shows how much of the variation in the dependent variable the independent variables describe together. R-squared is a convenient 0–100% scale that calculates the frequency of the interaction between your model and the dependent variable [25] . R-squared does not mean whether a regression model suits the data well. The R2 value of a good model can be negative. A skewed model, on the other hand, may have a high R2 value.

3.2.4 Adjusted R-Squared

R2 has the same definition as R2 but is an upgrade on it. R2 has the issue that the scores increase on increasing terms even if the model is not changing, which may lead the researcher astray. Adjusted R2 is always smaller than R2 so it accounts for rising predictors and only reveals change where there is one [24].

$$R_a^2 = 1 - \left[\!\left[\left(\frac{n-1}{n-k-1}\right) \times (1 - R^2) \right]\!\right]$$

Where n = number of observations, k = number of independent variables and R_a^2 = adjusted R2

Chapter 4: Analysis

4.1 Experimental Setup

The material used for this experimental setup is aluminium 1061 alloy with a dimension of length 380mm and a diameter of 38mm. Experiments were performed by machining Al-1061 using a high-speed steel tool, this experiment was conducted at the Engineering Development Institute Machining Workshop. The workpiece is clamped on both sides and after each turning operation, a scanning electron microscope (SEM) was used to measure the cutting tool [26].

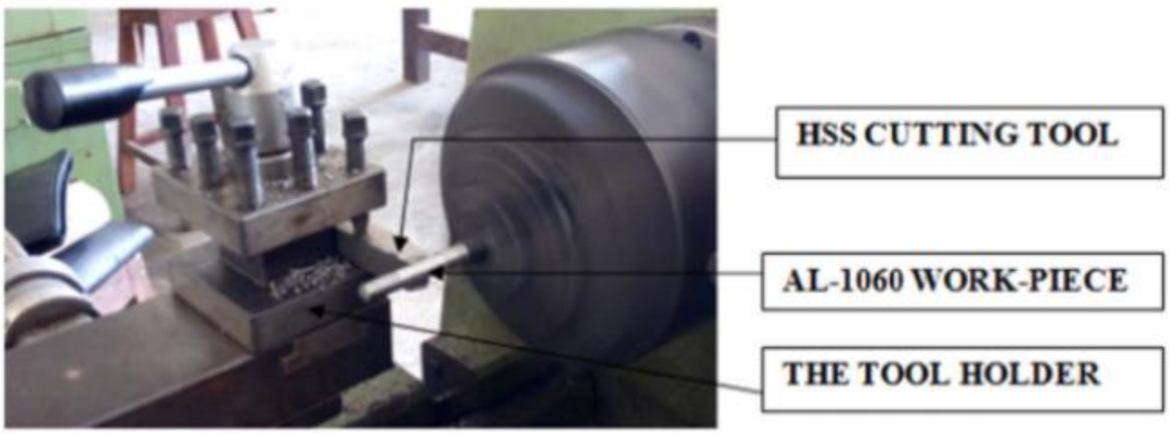

Figure 6

4.2 Data

Wear is an unwanted deterioration of an element when some quantity from the surface of the workshop is eliminated. it occurs when part of the working tool gets chipped off, this leads to tool deformation and then an increase in surface irregularity. In this data set, the process parameters such as feed rate, depth of cut and cutting speed do influence the product quality the maintenance costs. The experimental data shows the relationship in effect between cutting tool, workpiece, and machining parameters. it is useful in providing optimization parameters as it influences the nature of tool wear produced.

The dataset consists of 4 columns where Cutting Speed, Feed Rate and Radial Depth of Cut are the input parameters and TWmax is the output parameter that is supposed to be predicted. The entire dataset has 300 rows which means that it is relatively small for machine learning algorithms.

Table 3

SL NO	Cutting speed (rpm)	Feed rate (mm/min)	Radial depth of cut(mm)	Tool wearTWmax
1	150	50	1.5	0.226
2	150	100	1.5	0.286
3	150	150	1.5	0.26
4	150	50	1	0.293
5	150	100	1	0.237
6	150	150	1	0.296
7	150	50	0.5	0.272
8	150	100	0.5	0.299
9	150	150	0.5	0.34
10	200	50	1.5	0.409
11	200	100	1.5	0.379
12	200	150	1.5	0.424
13	200	50	1	0.36
14	200	100	1	0.415
15	200	150	1	0.386
16	200	50	0.5	0.221
17	200	100	0.5	0.312
18	200	150	0.5	0.363
19	250	50	1.5	0.316
20	250	100	1.5	0.359
21	250	150	1.5	0.311
22	250	50	1	0.343
23	250	100	1	0.213
24	250	150	1	0.445
25	250	50	0.5	0.324
26	250	100	0.5	0.33
27	250	150	0.5	0.321
28	150	50	1.5	0.226
29	150	100	1.5	0.286
30	150	150	1.5	0.26
31	150	50	1	0.293
32	150	100	1	0.237
33	150	150	1	0.296
34	150	50	0.5	0.272
35	150	100	0.5	0.299
36	150	150	0.5	0.34
37	200	50	1.5	0.409
38	200	100	1.5	0.379
39	200	150	1.5	0.424
40	200	50	1	0.36
41	200	100	1	0.415
42	200	150	1	0.386
43	200	50	0.5	0.221
44	200	100	0.5	0.312
45	200	150	0.5	0.363
46	250	50	1.5	0.316

Table 3 shows a sample subset of the dataset

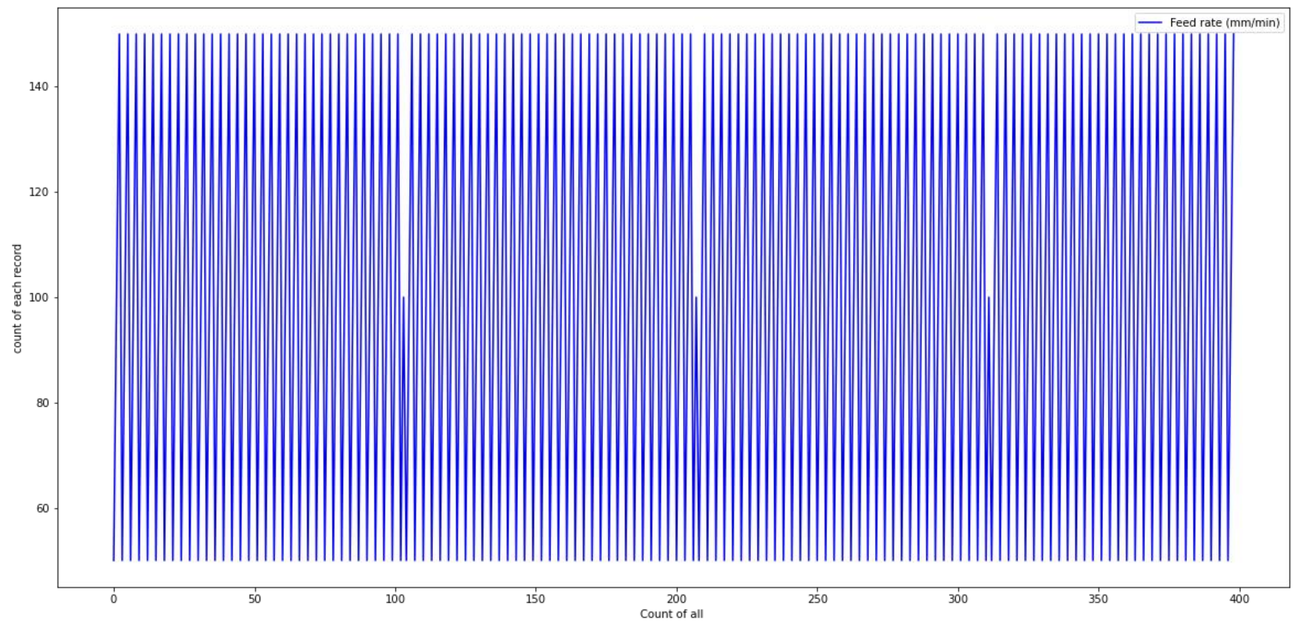

Figure 7

Figure 7 is a graph based on the reed rate that consists of the number of rows as the X-axis and the value of each row as the Y-axis.

Figure 8

Similarly, Figure 8 shows the radial depth of the cut.

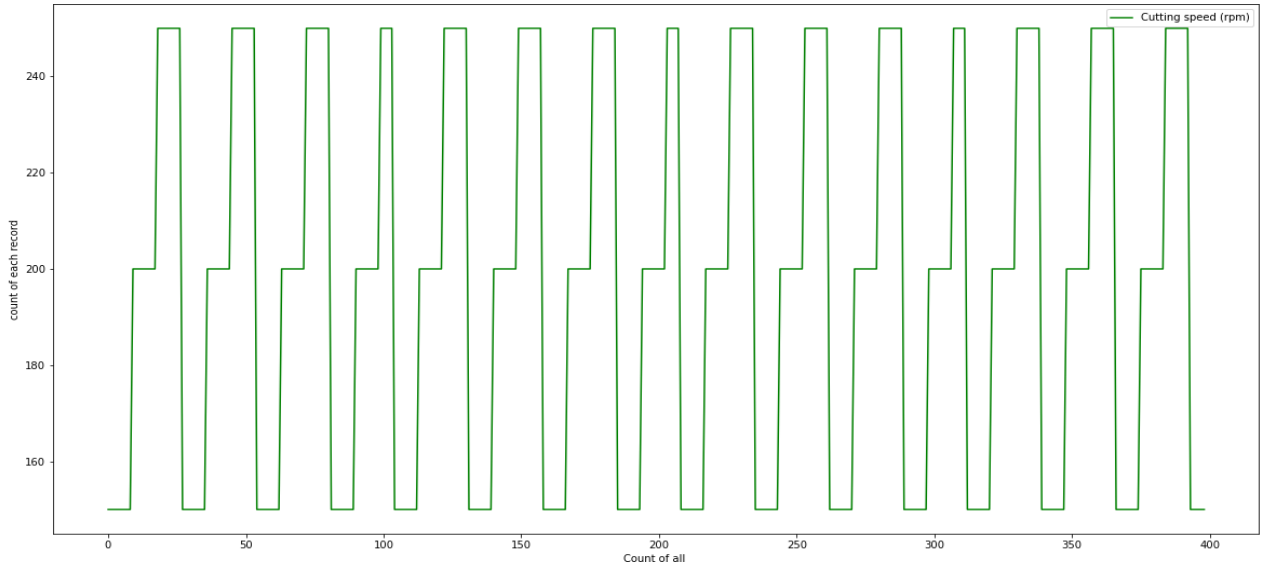

Figure 9

Figure 9 represents the cutting speed.

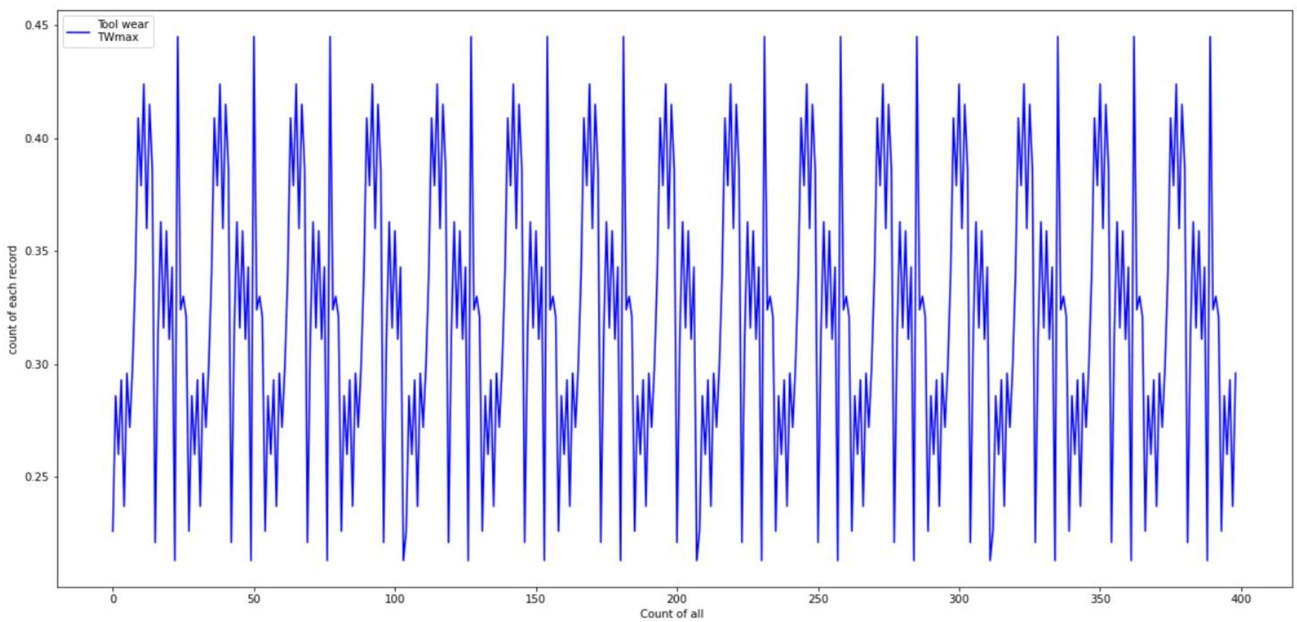

Figure 10

Figure 10 is the output parameter which is the tool wear. This needs to be predicted correctly to successfully implement a tool condition monitoring system.

4.2.1 Correlation

Correlation is a mathematical expression that refers to how close two variables are to forming a linear relationship with one another. It is preferable to reduce the number of input variables to reduce modelling computational costs and, in some cases, improve model performance.

High correlation features are more linearly dependent and thus have nearly the same effect on the dependent variable. When two features have a high correlation, we will remove one of them.

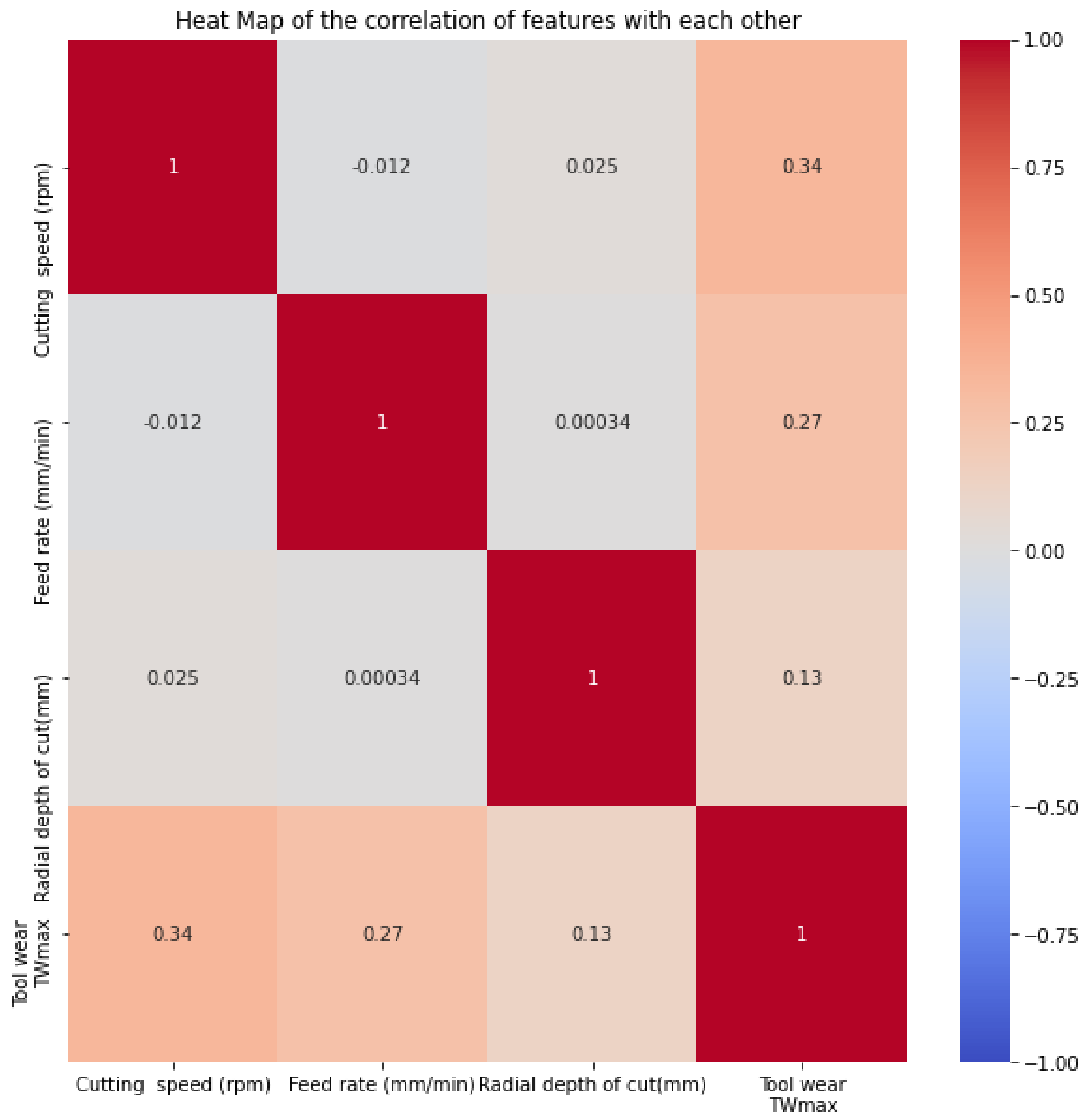

Figure 11

4.3 Pre-processing

There is no need for pre-processing as the raw data provides us with sufficient information and with feature vectors that could directly be put into a classifier to provide results, the data set consists of labelled data with no missing values. The dataset will be split up for training and testing the algorithm.

4.4 Cross-validation

Cross-validation was initially used to assess the statistical viability of linear regression models used to forecast a success criterion based on test scores. It was discovered that the various correlation coefficients within the initial study used to assign values to regression weights gave a positive impression of the statistical potency of the regression equation when applied to prospective findings [27]. Cross-validation was first used to investigate probability equations, but it has increasingly been used for model collection. Both applications of cross-validation include sample-based model tuning or the assigning of values to parameters.

Cross-validation is a resampling technique used to test machine learning models on a small sample of data.

The method has a single parameter called k that specifies the number of groups into which a given data sample should be divided. As a result, the technique is often referred to as k-fold cross-validation. When a new value for k is used, it could be used in place of k in the model's comparison, for example, k=10 resulting in 10-fold cross-validation.

It is a common approach since it is easy to grasp and produces a less skewed or positive estimation of model ability than other approaches, such as a simple train/test split.

4.4 Libraries used.

4.4.1 Pandas

Pandas is a data manipulation and research software library written for the Python programming language. It provides data structures and operations for controlling numerical tables and time series in particular. Pandas are very durable. This library gives the user a huge collection of commands and awesome functionality that can be used to easily analyse the given data.

Pandas have taken data mining to a whole new degree. It assists you in filtering data based on the parameters you have put in place, as well as segregating and segmenting your data based on your preferences.

4.4.2 scikit-learn.

scikit-learn provides a set of built-in machine learning algorithms and models it is open-source software that is accessible to everybody. it simplifies machine learning and has efficient tools for predictive data analysis, it is built on NumPy, SciPy and matplotlib [28]. It has tools for classification, regression, and clustering. it has features for pre-processing and can split up a data set to train and test different models. It is built on the programming language Python and is easy to use. It provides feature extraction and dimensionality reduction. It is the ideal tool for any machine learning project and is a high-level library as mentioned above is packed with different features making machine learning easier and more understandable.

Chapter 5 Results

Supervised Learning techniques

A program was written using the given dataset to try and predict the TWmax. The dataset was loaded using panda's library. Since all the values in the dataset were present and there were no missing values, we can assume that supervised algorithms would outperform the unsupervised algorithms. The data from the pandas was split into training and testing data with 75% of the data used for training. At times to avoid overfitting test size was as large as 50%. This was implemented when KFold cross-validation was not performing as expected.

5.1 Random Forest

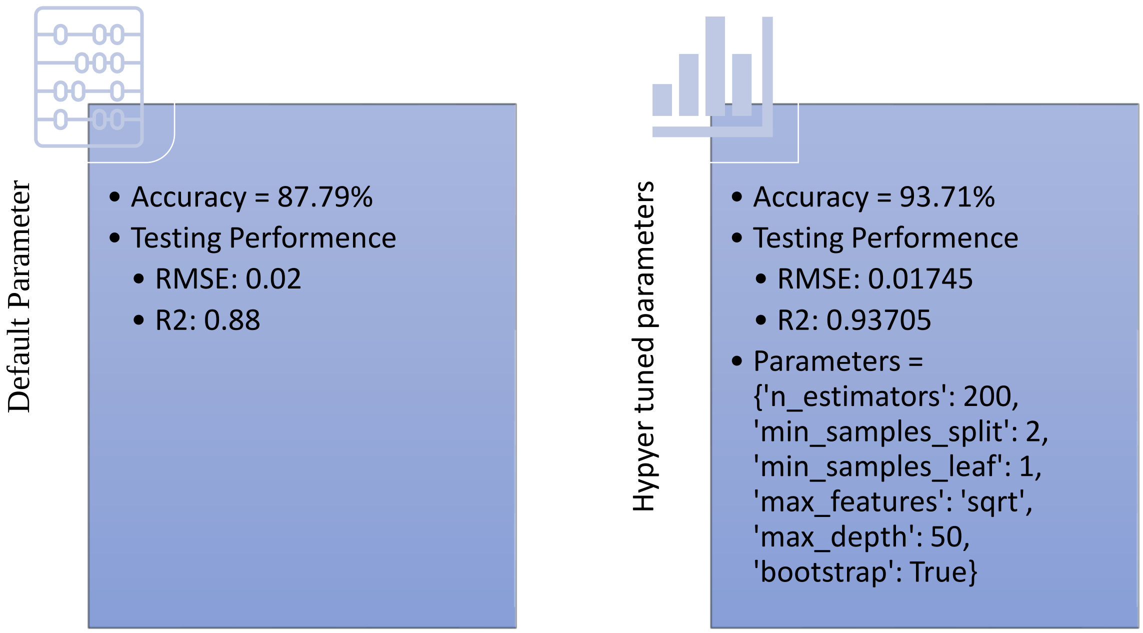

Figure 12

Hyper tunning the random forest parameters leads to an increase in the accuracy to 93%.

Figure 13 and Figure 14 are graphs showing the RF and hyper tuned RF predictions respectively.

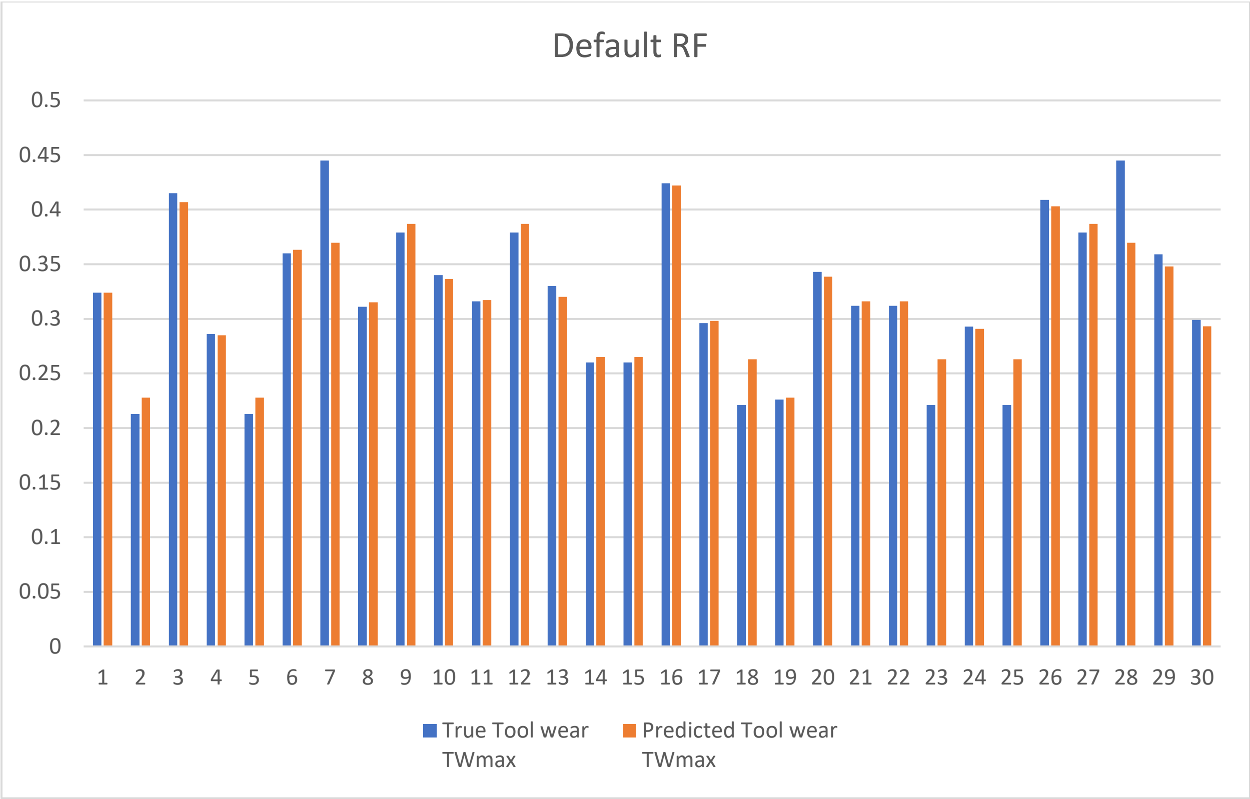

Figure 13

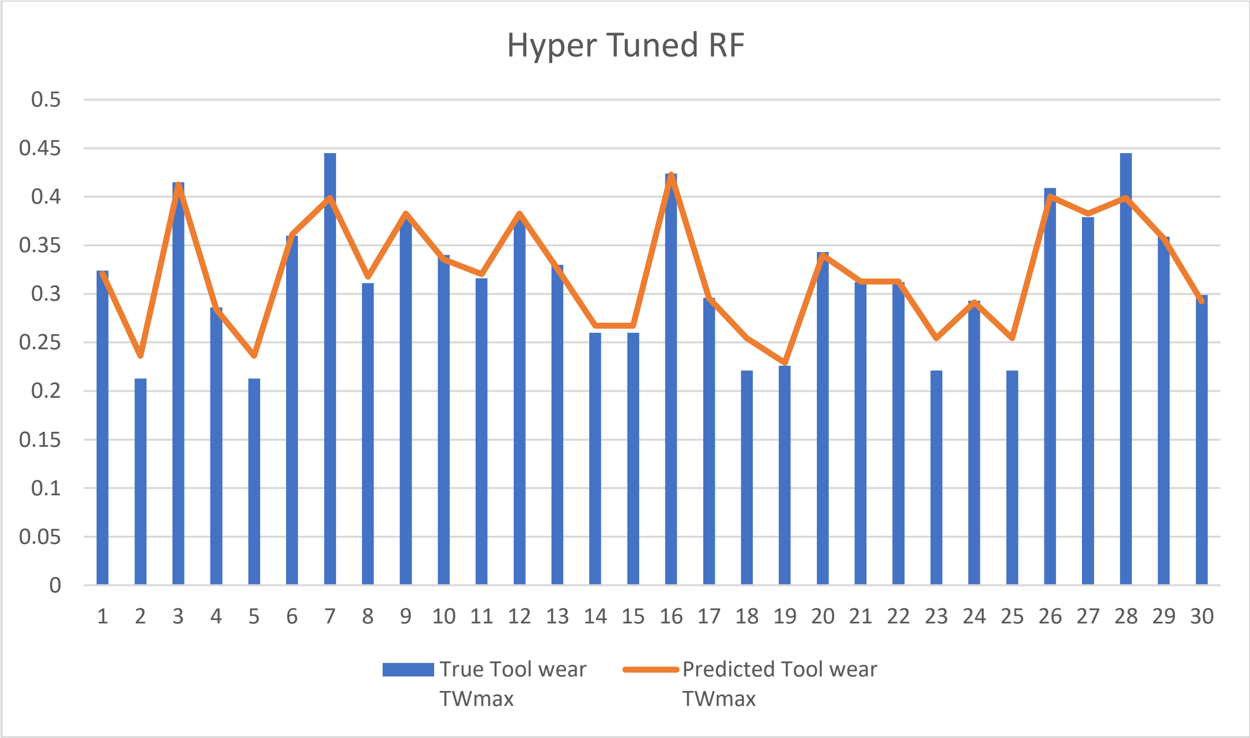

Figure 14

5.2 Support Vector Regressor

The SVR is a ML algorithm which is like K-means and has been used in this dataset. The results are discussed below. From Figure 15 it can be seen that SVR tends to predict the data with some accuracy where at times the value predicted is very far from the true value.

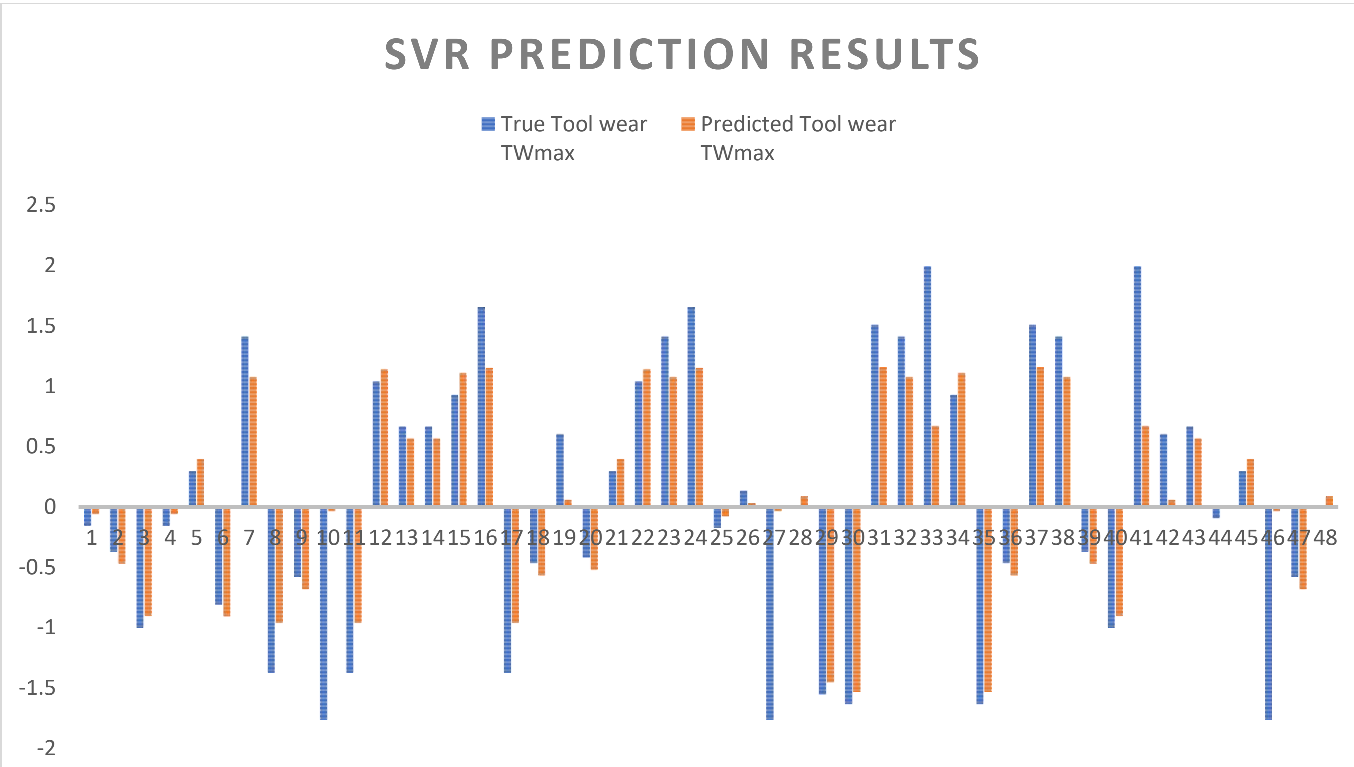

Figure 15

K_fold cross-validation of the dataset leads to a mean of 80.6% accuracy with a 5.88% standard deviation. This shows that the SVR algorithm does not overfit the dataset and preforms reasonably well on the given Dataset.

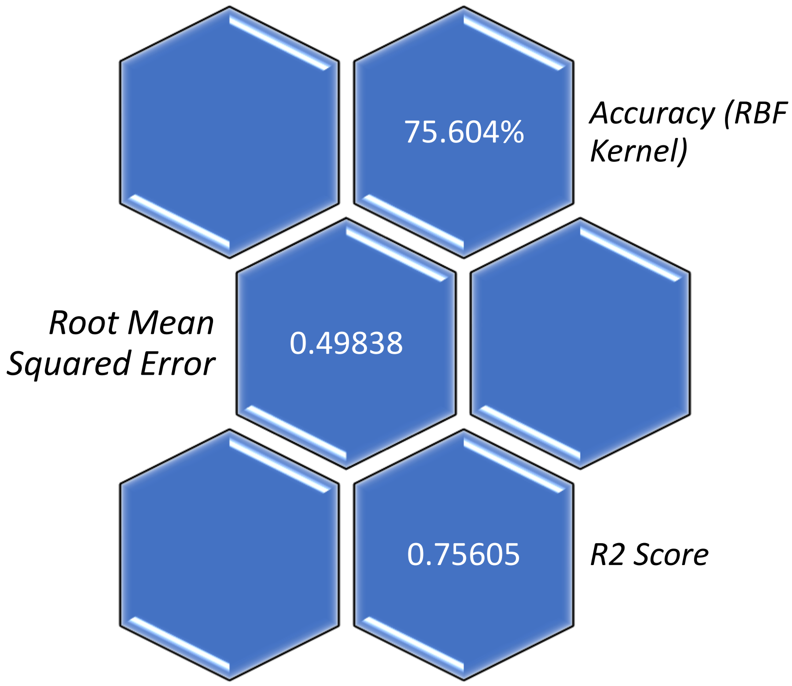

Figure 16

5.3 Ridge & Lasso Regression

Ridge & Lasso regression performed quite poorly on the data set. Hyper tunning the alpha parameter only made the results worse essentially making grid search not a viable option. All thought Ridge & Lasso regression have shown great results in other problems it fails to perform as required.

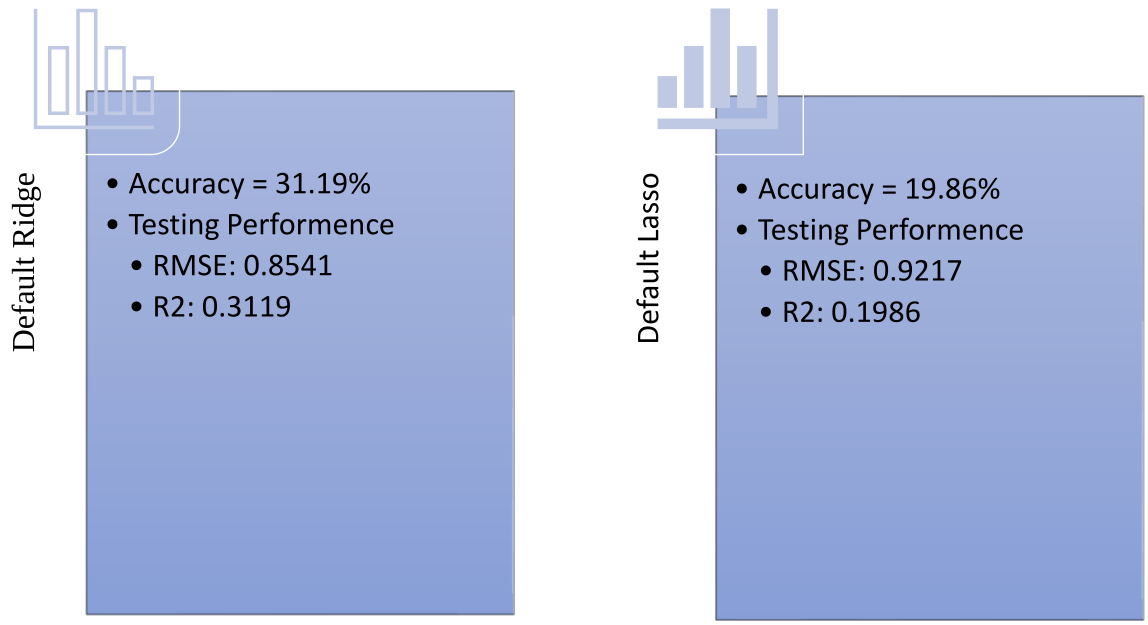

Figure 17

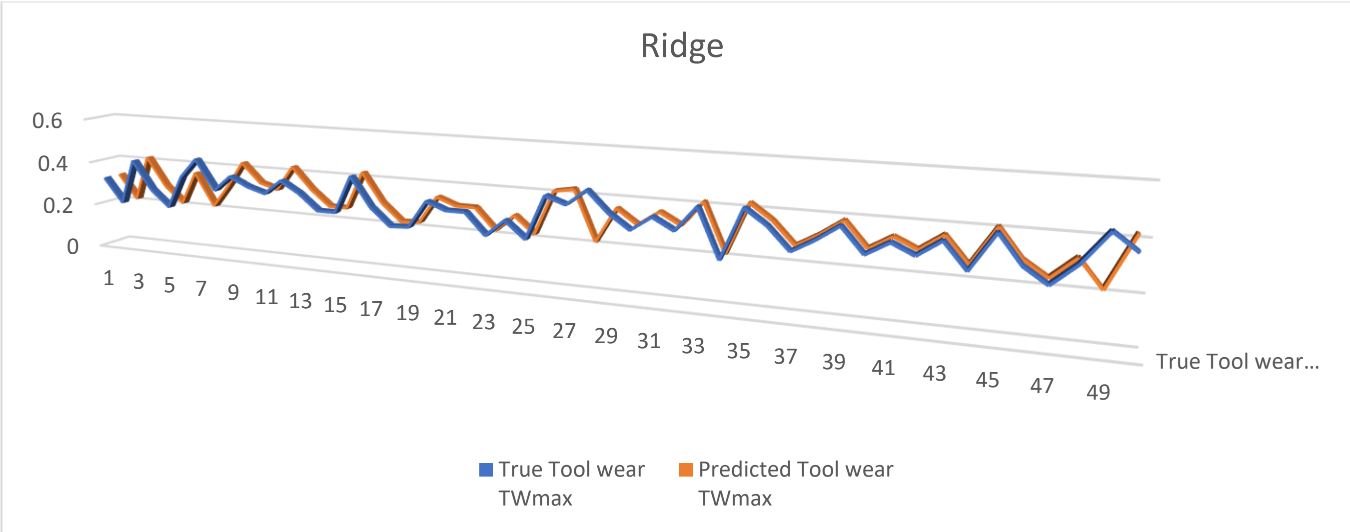

Figure 18

5.4 Boosting Algorithms

The following boosting method was implemented, and grid search was used to hyper tune GBM and Cat boost. These algorithms performed exceptionally well and could predict the tool wear with high accuracy.

Table 4

Algorithm	Parameters	Accuracy	RMSE	R2
GBM	learning_rate=0.01 max_depth=5 min_samples_split=3 n_estimators=600	0.8849	0.02155	0.88497
GBM	Default	0.3177	0.05249	0.31777
Cat Boost	loss_function:RMSE depth: 4 iterations: 150 learning_rate: 0.1 l2_leaf_reg: 0.2	0.9427	0.01529	0.94267

Cat Boost	Default	0.8780	0.02358	0.87796
XGB	Default	0.9631	0.01174	0.96314

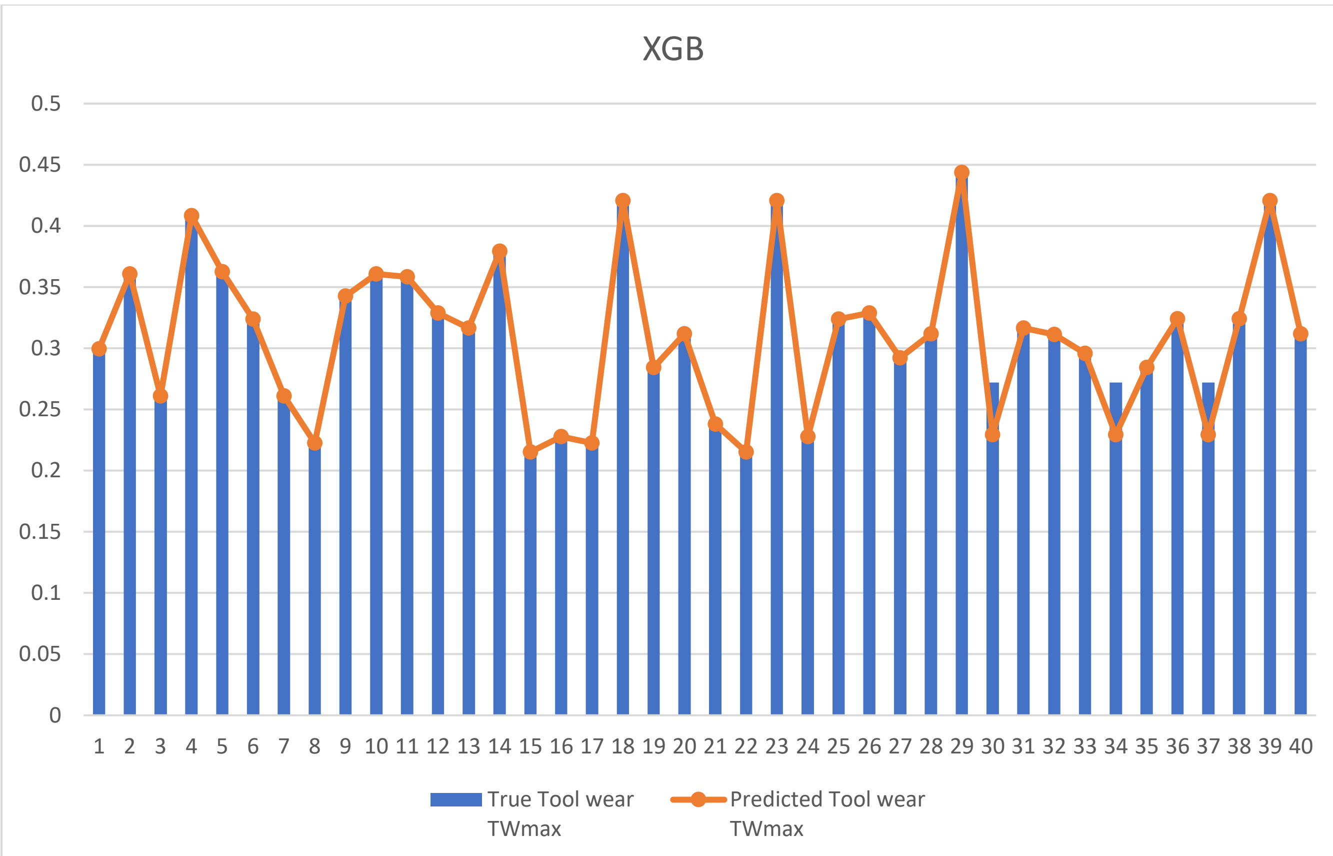

Figure 19

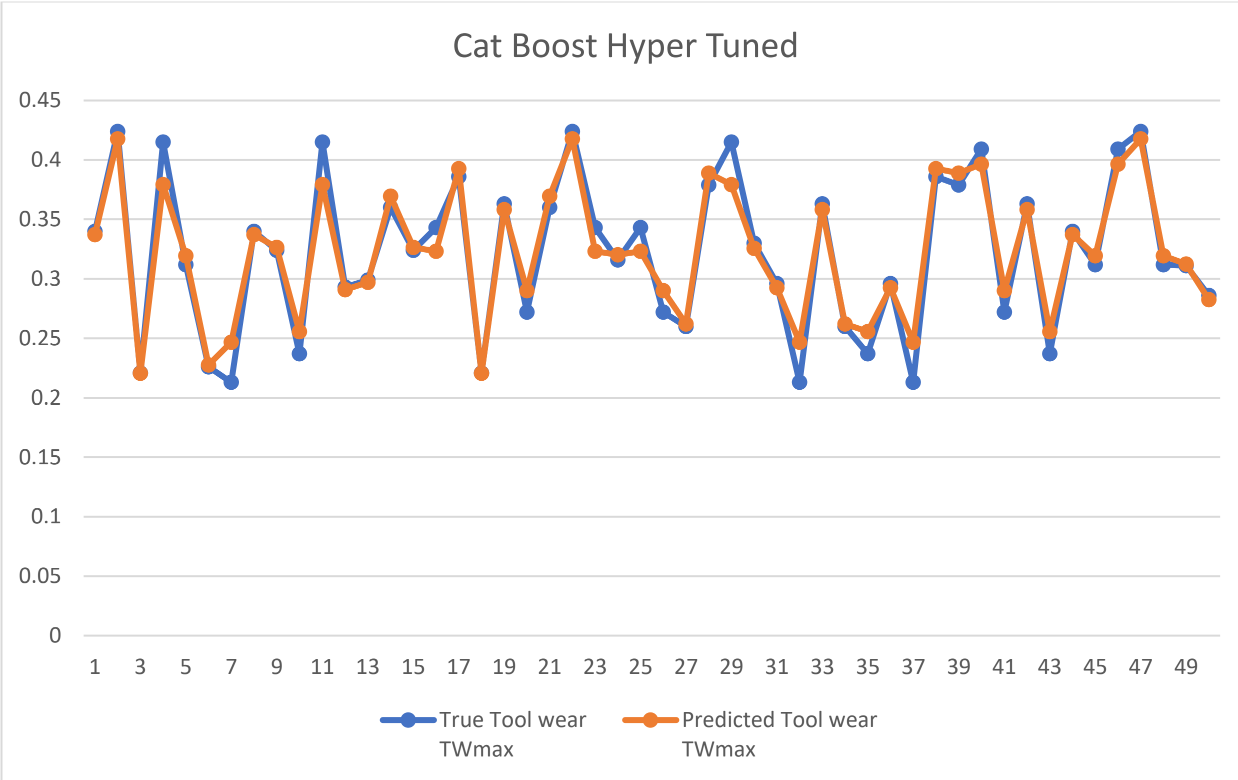

Figure 20

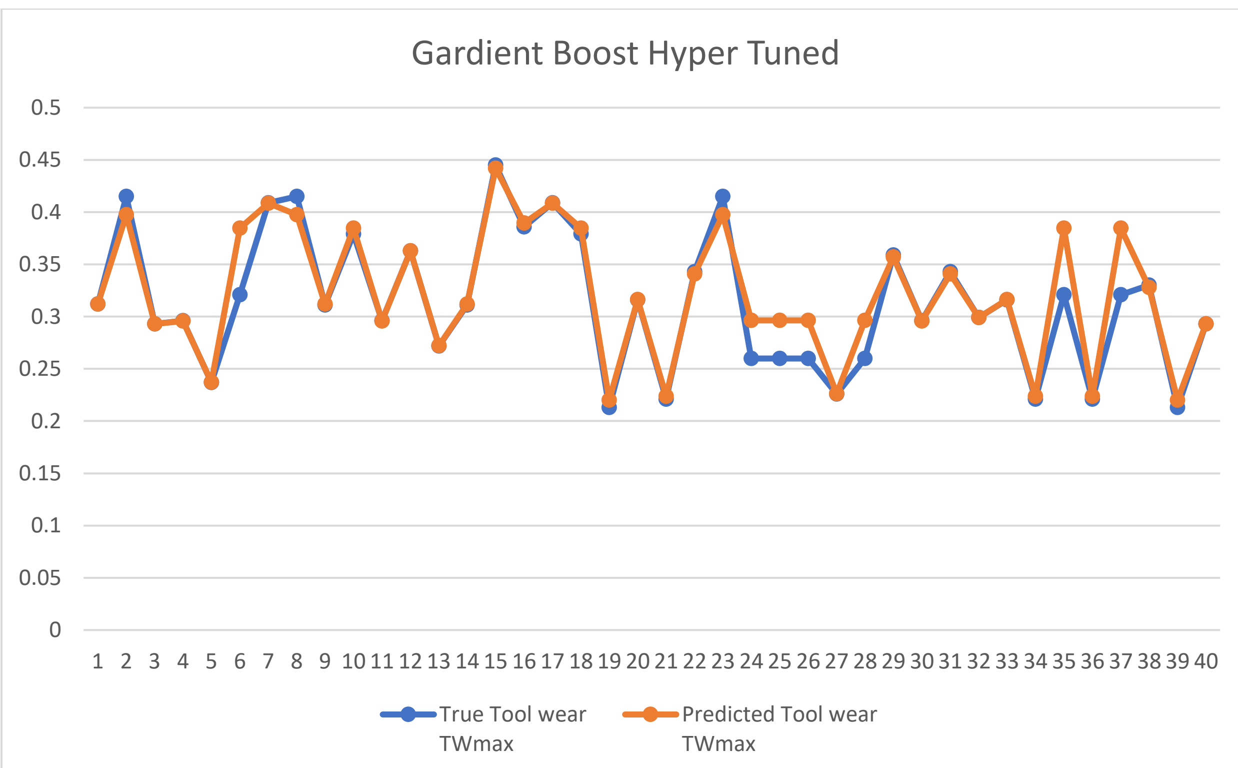

Figure 21

Chapter 6: Evaluation & Conclusion

6.1 Evaluation

The method to evaluate which algorithms should be selected is based on why how well the regressor is performing which can be calculated by the accuracy of the model. Different ML techniques were applied to the data set and here we summarize how each algorithm performed.

Ridge and Lasso's regression were the algorithms with the least accurate and the evaluation metrics results in Figure 17 support the fact that these had not performed as desired. An attempt was made to hyper tune the parameter alpha, but it resulted in the accuracy being even lower.

Support vector regressor takes a specific kernel as an input these include linear, poly, RBF, sigmoid, precomputed, after trying and testing with the all the kernels present the kernel which produced the is the default one which is RBF. SVR produced a much higher accuracy as shown in Figure 16 than Ridge and Lasso but in the cross-validation. There was no hyper tunning to the SVR as it did not show a significant increase in any of the evaluation metrics.

Tree-based algorithms such as random forest performed very well in the dataset with the default random forest giving 87% accuracy and the hyper tunning increased it another 5% Figure 12. KFold cross-validation was left out of the report as the mean was higher than expected. Overall random forest is one the best algorithms to implement for this regression problem.

The ensemble method is techniques that create multiple models and combine them to produce improved results. Gradient boosting is an example of an ensemble method. GBM without any hyper tunning has produced a very low accuracy but after using grid search it drastically improved the results. The accuracy went from 31% to 88% as shown in Table 4. Other boosting methods were implemented as GMB produced great results. Cat Boost is another type of boosting method where it has its library which was developed by catboost.ai. It provides a function where it shows how each input parameter affects the model. As shown in Figure 22. Another boosting method is XGB this algorithm did not need any hyper parameter tuning as the default model produced an accuracy of 96% as shown in Table 4. It also has the lowest RMSE values and the highest R2 of almost 1.

The final algorithm which was chosen to predict the tool condition was the average of the results hyper tunned random forest and the XDG Boosting. This produces an accuracy of 97% with testing performances RMSE: 0.0093757 & R2: 0.9765049. Figure 23 is the graph of the model which was made from Table 4 from Appendix A.

6.2 Conclusion

- ML techniques can predict tool wear very efficiently and the goal of tool condition monitoring was achieved through experimenting with different ML techniques and combing the different models.
- With Industry 4.0 being the new norms for the manufacturing industry there is a high demand to implement ML in different aspects to help and automate factories and monitor them efficiently.
- The correlation of input parameters shows that it was necessary to keep all the input parameters.
- Unsupervised learning algorithms were disregarded as all the data was labelled and there was no missing data.
- Initially, this was thought to be a classification problem, but it was a regression problem where the model had to predict a number instead of a class labelled as tool wear
- Ensemble methods such as GBM, XGB and Cat boosting are a few of the algorithms which performed the best followed by RF.
- The average of GBM and RF models produced the most accurate model.
- The evaluation metrics were closely selected such that a classifier did not only depend on the accuracy of the scikit-learns library. RMSE and R2 helped in identifying the best algorithms.
- Hyper tunning the parameters was critical as it improved the accuracy and other model metrics by a significant amount.
- Cross-validation was used to determine if the ML techniques were overfitting the model and it worked with a few limitations.
- The main goal of this project was to develop a classifier that would identify and prevent unplanned downtime of the machinery. This has been achieved as tool wear was predicted and this model could be used in real-time to check the efficiency.

6.3 Further Discussion and Future work

Further work could be done by introducing neural networks and other deep learning techniques. There could have also been an attempt made to work with the machinery directly to test the models on real-time data. Overall, the model produced is not perfect and could be overfitting the data, for future work a different dataset or a dataset with more data would allow better training and testing. Cross-validation was limited as KFold cross-validation gave perfect accuracy when testing boosting and random forest regressors. Measures were taken to reduce overfitting by splitting the dataset into 40-50% testing data in which the results above were produced. There could have been an overall improvement in the efficiency and quality of work if time were more efficiently used. Having a standalone algorithm seems redundant and there could also be a possibility of the model would not perform as expected and inform the operator that the tool needs to change. This could end increasing the overhead costs of the products produced from the tool being monitored.

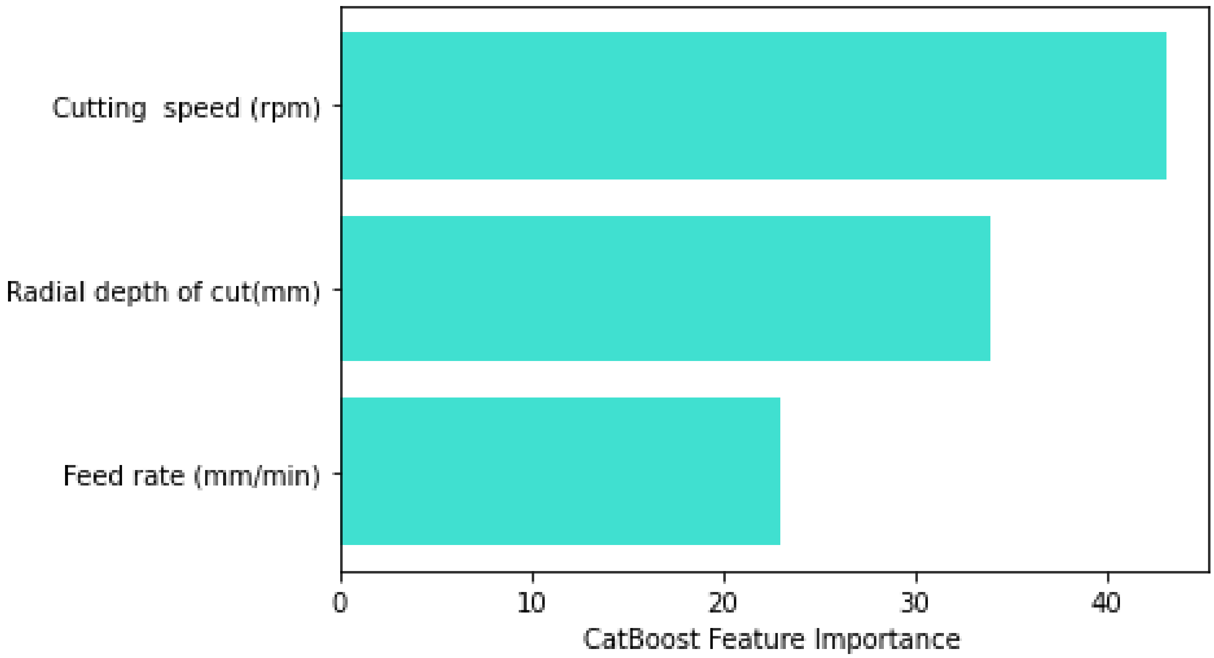

Figure 22

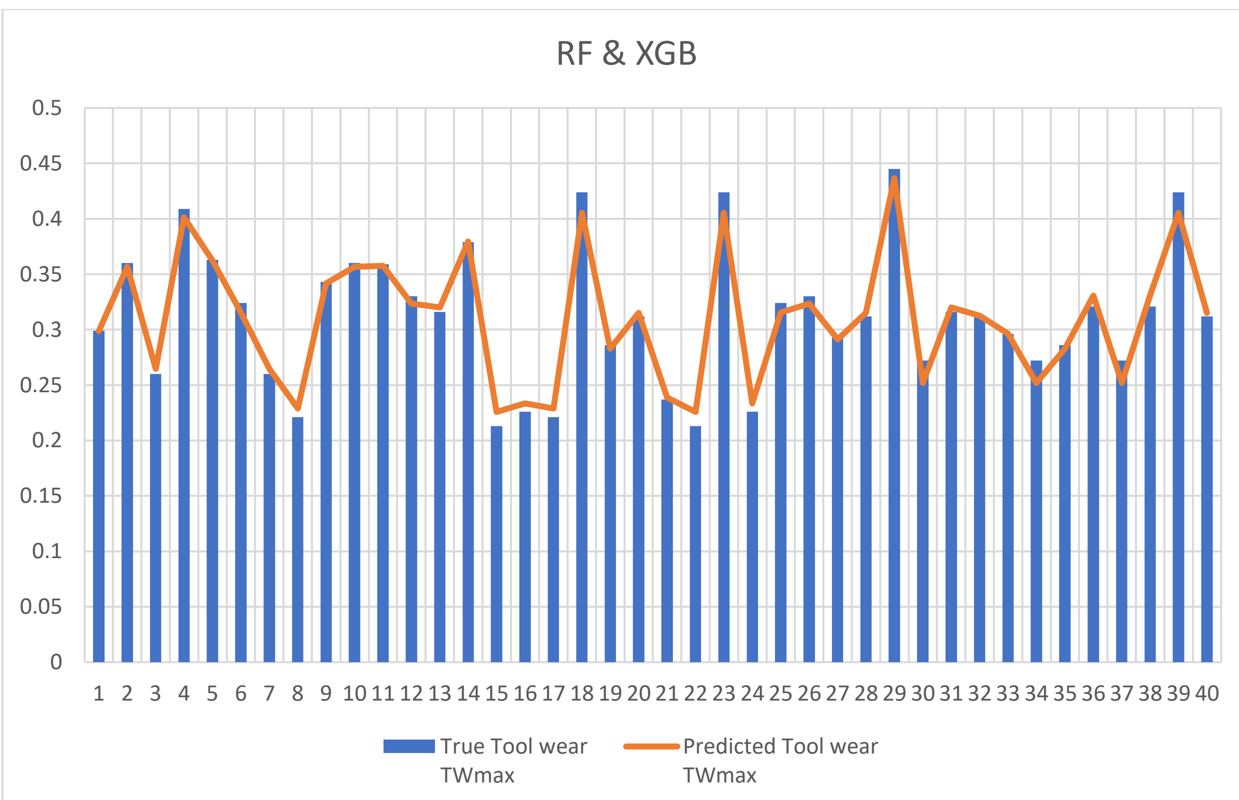

Figure 23

7 Appendix A: Regression Model Predictions

Table 1: Cat Boost

True Tool wear TWmax	Predicted Tool wear TWmax	Std
0.34	0.33716	0.002008
0.424	0.417609	0.004519
0.221	0.220587	0.000292
0.415	0.379275	0.025261
0.312	0.31955	0.005339
0.226	0.227508	0.001067
0.213	0.246674	0.023811
0.34	0.33716	0.002008
0.324	0.326392	0.001691
0.237	0.255533	0.013105
0.415	0.379275	0.025261
0.293	0.290777	0.001572
0.299	0.297195	0.001276
0.36	0.369473	0.006698
0.324	0.326392	0.001691
0.343	0.323147	0.014038
0.386	0.392557	0.004637
0.221	0.220587	0.000292
0.363	0.358245	0.003362
0.272	0.290027	0.012747
0.36	0.369473	0.006698
0.424	0.417609	0.004519
0.343	0.323147	0.014038
0.316	0.320304	0.003043
0.343	0.323147	0.014038
0.272	0.290027	0.012747
0.26	0.262097	0.001483
0.379	0.389036	0.007097
0.415	0.379275	0.025261
0.33	0.325601	0.00311
0.296	0.292485	0.002485
0.213	0.246674	0.023811
0.363	0.358245	0.003362
0.26	0.262097	0.001483
0.237	0.255533	0.013105
0.296	0.292485	0.002485
0.213	0.246674	0.023811
0.386	0.392557	0.004637

Table 2: XGB Boost

True Tool wear TWmax	Predicted Tool wear TWmax	Std
0.312	0.311879	8.54E-05
0.415	0.397683	0.01224504
0.293	0.293084	5.91E-05
0.296	0.295903	6.84E-05
0.237	0.237371	0.00026199
0.321	0.384553	0.04493889
0.409	0.408709	0.00020569
0.415	0.397683	0.01224504
0.311	0.311956	0.0006762
0.379	0.384767	0.00407802
0.296	0.295903	6.84E-05
0.363	0.362915	6.00E-05
0.272	0.272134	9.48E-05
0.311	0.311956	0.0006762
0.445	0.44207	0.00207184
0.386	0.389608	0.00255097
0.409	0.408709	0.00020569
0.379	0.384767	0.00407802
0.213	0.220082	0.00500798
0.316	0.316268	0.00018931
0.221	0.223901	0.00205124
0.343	0.340826	0.00153738
0.415	0.397683	0.01224504
0.26	0.296501	0.02581041
0.26	0.296501	0.02581041
0.26	0.296501	0.02581041
0.226	0.226245	0.00017301
0.26	0.296501	0.02581041
0.359	0.357217	0.0012607
0.296	0.295903	6.84E-05
0.343	0.340826	0.00153738
0.299	0.299069	4.89E-05

Table 3: GBM

True Tool wear TWmax	Predicted Tool wear TWmax	Std
0.312	0.311879	8.54E-05
0.415	0.397683	0.01224504
0.293	0.293084	5.91E-05
0.296	0.295903	6.84E-05
0.237	0.237371	0.00026199
0.321	0.384553	0.04493889
0.409	0.408709	0.00020569
0.415	0.397683	0.01224504
0.311	0.311956	0.0006762
0.379	0.384767	0.00407802
0.296	0.295903	6.84E-05
0.363	0.362915	6.00E-05
0.272	0.272134	9.48E-05
0.311	0.311956	0.0006762
0.445	0.44207	0.00207184
0.386	0.389608	0.00255097
0.409	0.408709	0.00020569
0.379	0.384767	0.00407802
0.213	0.220082	0.00500798
0.316	0.316268	0.00018931
0.221	0.223901	0.00205124
0.343	0.340826	0.00153738
0.415	0.397683	0.01224504
0.26	0.296501	0.02581041
0.26	0.296501	0.02581041
0.26	0.296501	0.02581041
0.226	0.226245	0.00017301
0.26	0.296501	0.02581041
0.359	0.357217	0.0012607
0.296	0.295903	6.84E-05
0.343	0.340826	0.00153738
0.299	0.299069	4.89E-05
0.316	0.316268	0.00018931
0.221	0.223901	0.00205124
0.321	0.384553	0.04493889
0.221	0.223901	0.00205124
0.321	0.384553	0.04493889
0.33	0.327867	0.00150828
0.213	0.220082	0.00500798

Table 4: XGB & RF avg

True Tool wear TWmax	Predicted Tool wear TWmax	Std
0.299	0.298620821	0.000268
0.36	0.356646818	0.002371
0.26	0.264740174	0.003352
0.409	0.401333443	0.005421
0.363	0.362465272	0.000378
0.324	0.315403109	0.006079
0.26	0.264740174	0.003352
0.221	0.228877398	0.00557
0.343	0.341521169	0.001046
0.36	0.356646818	0.002371
0.359	0.357420804	0.001117
0.33	0.32336638	0.004691
0.316	0.320144648	0.002931
0.379	0.37934649	0.000245
0.213	0.225759059	0.009022
0.226	0.233326456	0.005181
0.221	0.228877398	0.00557
0.424	0.405425994	0.013134
0.286	0.28288252	0.002204
0.312	0.315011869	0.00213
0.237	0.238779731	0.001258
0.213	0.225759059	0.009022
0.424	0.405425994	0.013134
0.226	0.233326456	0.005181
0.324	0.315403109	0.006079
0.33	0.32336638	0.004691
0.293	0.291259103	0.001231
0.312	0.315011869	0.00213
0.445	0.436637623	0.005913
0.272	0.251915425	0.014202
0.316	0.320144648	0.002931
0.311	0.312628727	0.001152
0.296	0.296247047	0.000175
0.272	0.251915425	0.014202
0.286	0.28288252	0.002204
0.321	0.330665327	0.006834
0.272	0.251915425	0.014202
0.321	0.330665327	0.006834
0.424	0.405425994	0.013134

Table 4: Random Forest hyper tunned

True Tool wear TWmax	Predicted Tool wear TWmax
-1.630062944	-0.114462723
-0.099865516	0.171270687
-1.372345482	-0.245468274
-0.808588535	-0.395262709
0.608857503	-0.111996011
-0.808588535	-0.395262709
0.028993214	0.166337263
-1.758921674	0.316131698
-1.549526237	-0.390329285
-0.422012343	-0.09814055
1.027648378	0.182659436
0.125637262	0.313664986
-0.164294882	0.032865
1.639727348	0.185126148
-0.583085757	-0.243001562
-0.01932881	0.46099271
1.977981517	0.463459422
0.657179527	0.180192724
1.027648378	0.182659436
-0.01932881	0.46099271

Table 5: SVR

True Tool wear TWmax	Predicted Tool wear TWmax
-0.160349444	-0.059985192
-0.37091465	-0.470780959
-1.00261027	-0.902381406
-0.160349444	-0.059985192
0.293175616	0.393600014
-0.808242387	-0.907788248
1.410790943	1.075399073
-1.375148712	-0.962539126
-0.581479857	-0.681822677
-1.763884478	-0.035691336
-1.375148712	-0.962539126
1.038252501	1.138584667
0.665714058	0.56560467
0.665714058	0.56560467

0.600924764	0.057572026
0.600924764	0.057572026
-0.176546767	-0.076862643
1.038252501	1.138584667
0.341767587	0.241320553
-0.014573532	0.085771297
1.410790943	1.075399073
-0.09556015	0.004845754
0.034018439	-0.065763155
0.665714058	0.56560467
-0.468098592	-0.568405854
-0.014573532	0.085771297
0.617122088	0.717252271
-0.581479857	-0.681822677
1.653750797	1.149755113
-0.160349444	-0.059985192
-0.160349444	-0.059985192
-0.37091465	-0.470780959
-1.553319271	-1.453428494
0.617122088	0.717252271
0.665714058	0.56560467
-0.09556015	0.004845754
-0.176546767	-0.076862643
0.665714058	0.56560467
-1.553319271	-1.453428494
0.293175616	0.393600014
-0.37091465	-0.470780959
0.293175616	0.393600014
1.993894592	0.66994839
0.600924764	0.057572026
-1.375148712	-0.962539126
0.924871236	1.111126909
-0.37091465	-0.470780959
0.293175616	0.393600014
1.993894592	0.66994839
0.034018439	-0.065763155
-1.553319271	-1.453428494
-0.419506621	-0.519457106
0.600924764	0.057572026
1.653750797	1.149755113
-0.176546767	-0.076862643
0.13120238	0.030949818
-1.375148712	-0.962539126

8 Appendix B: External Libraries

There were several libraries used apart from the ones present in the anaconda library. Here is how you could install the following.

First go the anaconda prompt where pip is installed. Then type in the following commands to have cat boost and xgb boost libraries installed.

```
➢ pip install xgboost
➢ pip install catboost
```

9 References

[1] X. E. K. S. T. Ray Y.Zhonga, “Intelligent Manufacturing in the Context of Industry 4.0: A Review,” *Engineering,* vol. 3, no. 5, pp. 616-630, 2017.

[2] L. D. F. A. Alejandro Germán Frank, “Industry 4.0 technologies: Implementation patterns in manufacturing companies,” *International Journal of Production Economics,* vol. 210, pp. 15-26, 2019.

[3] G. P. P. C. Chiara Cimini, “Industry 4.0 Technologies Impacts in the Manufacturing and Supply Chain Landscape: An Overview: Proceedings of SOHOMA 2018,” in *Service Orientation in Holonic and Multi-Agent Manufacturing*, Springer-Verlag Berlin Heidelberg, 2019, pp. 109-120.

[4] P. D. F. C. K. Dr. Andreas H. Glas, “The Impact of Industry 4.0 on Procurement and Supply Management: A Conceptual and Qualitative Analysis,” *International Journal of Business and Management Invention,* vol. 5, no. 6, pp. 55-66, 2016.

[5] H. H. J. H. A. W. W. Kagermann, “Recommendations for implementing the strategic initiative INDUSTRIE 4.0:,” acatech, Wirthschaft, 2013.

[6] N. Ambhore, “Tool Condition Monitoring System: A Review,” pp. 2-3, 2015.

[7] R. K. Mobley, An Introduction to Predictive Maintenance, Elsevier Science, 2002.

[8] GE, “Predix Asset Performance Management,” GE, [Online]. Available: https://www.ge.com/digital/applications/asset-performance-management. [Accessed 15 December 2020].

[9] D. T. Pham, “Machine-learning techniques and their applications in manufacturing,” *Journal of engineering manufacture,* vol. 209, pp. 395-412, 2016.

[10] T. Wuest, “Machine learning in manufacturing: advantages, challenges, and applications,” *Production & Manufacturing Research,* vol. 4, no. 1, 2016.

[11] S. KOTSIANTIS, Supervised Machine Learning: A review of Classification Techniques, IOS Press, 2007.

[12] IBM Cloud Education, “Supervised Learning,” IBM, [Online]. Available: https://www.ibm.com/cloud/learn/supervised-learning#toc-how-superv-A-QjXQz-. [Accessed 15 December 2020].

[13] J. B. W. a. D. H. F. G. Biswas, “ITERATE: a conceptual clustering algorithm for data mining,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C,* vol. 28, pp. 219-230, 1998.

[14] B.-S. Y. Achmad Widodo, “Support vector machine in machine condition monitoring and fault diagnosis,” *Mechanical Systems and Signal Processing,* vol. 21, no. 6, pp. 2560-2574, 2007.

[15] C. Piech, “K Means,” 2013. [Online]. Available: https://stanford.edu/~cpiech/cs221/handouts/kmeans.html. [Accessed 15 December 2020].

[16] D. P. R. M. Trupti M. Kodinariya, “Review on determining number of Cluster in,” *International Journal of Advance Research in Computer Science and Management Studies,* vol. 1, no. 26, 2013.

[17] T. Wood, “Random Forests,” [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/random-forest. [Accessed 25 December 2020].

[18] N. Donges, “A COMPLETE GUIDE TO THE RANDOM FOREST ALGORITHM,” 3 September 2020. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm. [Accessed 25 December 2020].

[19] s.-l. developers, “sklearn.ensemble.RandomForestClassifie,” [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html. [Accessed 27 December 2020].

[20] C. Hans, “Bayesian lasso regression,” *Biometrika,* vol. 96, no. 4, pp. 835-845, 2009.

[21] J. H.Friedman, “Stochastic gradient boosting,” *Computational Statistics & Data Analysis,* vol. 38, no. 4, pp. 367-378, 2002.

[22] T. H. R. T. Jerome Friedman, “Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors),” *Annals of statistics,* vol. 28, no. 2, pp. 337-407, 2000.

[23] S. H. Ather, “How to Calculate RMSD,” 28 12 2020. [Online]. Available: https://sciencing.com/calculate-rmsd-5146965.html. [Accessed 5 5 2021].

[24] D. Mishra, “Regression: An Explanation of Regression Metrics And What Can Go Wrong,” 7 12 2019. [Online]. Available: https://towardsdatascience.com/regression-an-explanation-of-regression-metrics-and-what-can-go-wrong-a39a9793d914.

[25] J. Frost, “How To Interpret R-squared in Regression Analysis,” 2020. [Online]. Available: https://statisticsbyjim.com/regression/interpret-r-squared-regression/.

[26] O. O. C. B. ,. K. O. I.P. Okokpujie, “Experimental data-set for prediction of tool wear,” *Data in Brief,* pp. 1196-1203, 2018.

[27] M. W. Browne, “Cross-Validation Methods,” *Journal of Mathematical Psychology,* vol. 44, no. 1, pp. 108-132, 2000.

[28] “scikit-learn,” [Online]. Available: https://scikit-learn.org/stable/. [Accessed 15 December 2020].

[29] GE, “Vale Fertilizantes Saves $1.4M in Production Losses with Predix Asset Performance Management,” GE, [Online]. Available: https://www.ge.com/digital/customers/vale-fertilizantes-saves-14m-production-losses-predix-asset-performance-management#to-section-index=section-3. [Accessed 15 December 2020].