12-2022

Predicting Failure Rate of Oil & Gas Equipment Using ML

Shaima Alblooshi
sha4435@g.rit.edu

RIT

Predicting Failure Rate of Oil & Gas Equipment Using ML

by

Shaima Alblooshi

A Capstone Submitted in Partial Fulfilment of the Requirements for the Degree of Master of Science in Professional Studies:

Data Analytics

Department of Graduate Programs & Research

Rochester Institute of Technology

RIT Dubai

DEC 2022

RIT

Master of Science in Professional Studies:

Data Analytics

Graduate Capstone Approval

Student Name: Shaima Alblooshi

Graduate Capstone Title: Predicting Failure Rate of Oil & Gas Equipment Using ML

Graduate Capstone Committee:

Name: Dr. Sanjay Modak Date:

Chair of committee

Name: Dr. Ehsan Warriach Date:

Member of committee

ABSTRACT

Value of time has become an important perspective in business application ranging from day to day working to big businesses. The value of time is more important in the case of refinery business which has become of paramount importance with increasing energy needs. The main point of contention in refinery operations is the periodic maintenance of the pipelines which consumes of valuable time and resources. With a proper solution which can cater the time requirements of the lead time. The fact is that time consumption is extremely critical for the operations of refinery. Therefore, the application of machine learning is implemented in the prediction of when and how the equipment will be needing preventive maintenance all of this can be accomplished by using available open-source data which will help us in the designing the algorithm and also in the learning of the same. This model has allowed us to investigate different outcomes and planning strategies that are possible through the prediction models and the estimated timings for the maintenance of the pipelines. This predictive maintenance system has allowed for more intelligent and smart planning and has reduced the down time significantly allowing for more revenues.

Keywords: Machine Failure, predictive maintenance, preventive maintenance, failures, machine learning, equipment.

CONTENTS

CHAPTER 1

1.1 Background of the Problem

Any predetermined plan for preventative maintenance is often costly and time-consuming given that systems and their isolated modules that are set for preventative maintenance are normally in good condition and could be used for lengthy periods. There are numerous issues based on the scheduled preventative maintenance that result in delays in product delivery and revenue loss due to delays associated with too much time spent on the process. For example, when a refinery and pipeline that are producing and delivering the product are stopped from working and have to be checked for any issue, the entire process plus the time to re-energize the entire system (interrelated parts) will require a lot of money. Research shows that with these fixed schedules, the maximum optimization of the system is never achieved. If preventive maintenance is not carried out on time, it will affect the safety and health protocols that have been established and ignoring them will lead to a catastrophe that will have severe effects on the environment and human life. As a result, this is a two-edged problem that requires a proper, long-term solution that can prevent the following: loss of manpower and revenue, delays in supplies, and lack of adherence to safety and health protocols. Thus, the current project focuses on the best analytics or predictive model to address the difficulties that oil and gas companies encounter during their turn-around and preventive maintenance schedules for their equipment.

1.2 Statement of problem

As preventative maintenance is becoming costly to operational processes, wastes resources, and takes up valuable time, many oil and gas companies are currently having trouble defining the schedules to carry out the process in a timely and cost-effective manner. Therefore, the issue at hand is to come up with a solution for an overall maintenance system that can be used by oil and gas companies for their equipment to reduce time and, most significantly, cut costs, by turning preventive maintenance into predictive maintenance to lower both time and cost.

1.3 Project Definitions and Goals

The definition of the project is to design and implement an artificial intelligence (AI) based system which can detect the failure of equipment beforehand and make appropriate adjustments in deciding when and how to initiate the preventive maintenance. Furthermore, this project is looking to achieve specific goals, which will also be the deciding factor of the project's overall success; these key performance indicators or goals are as follows.

1. Creating a system that will reduce the staffing as much as possible by precisely pinpointing which equipment and assembly needs maintenance; this decision-making process will ensure that the minimum workforce is utilized and achieve higher throughput form.
2. Able to reduce the cost of the maintenance and schedule and predict maintenance before time
3. Increase availability of the system, thus increasing the throughput, which will eventually increase the revenues.
4. by Providing the proper training of the algorithm and constant feedback and upgrades; the proposed system could be matured enough to detect any failure before it happens and adequately plan out the maintenance activity
5. The final goal is to design and deploy a system that will transfer the scheduled preventive maintenance into a coordinated predictive maintenance system.

1.4 Project Deliverables

Following are the projects deliverables

1. A solution for preventive maintenance
2. transfer the scheduled preventive maintenance into a coordinated predictive maintenance system
3. Build a Proper predictive maintenance model
4. Appropriate prediction of the impending errors, based on the preconditions
5. Development of datasets for further improvement of the system
6. ML model that can best predicts the failure of oil and gas equipment and a comparison with different ML models

CHAPTER 2

2.1 Literature Review

Machine learning (ML) is a tool that has many uses and can be implemented for both predictive maintenance and equipment optimization purposes. According to Montero Jimenez et al. (2020)[15], predictive maintenance scenarios involve data being collected, in most cases over time, to monitor the steady-state and efficiency of equipment and machinery within a more extensive system. The purpose of predictive maintenance is to find trends and patterns that can be used to predict and, eventually, prevent equipment malfunctions and failures (Carvalho et al., 2019)[7]. According to Abidi et al. (2022)[1], each manufacturing domain integrates computers and digitalization into its operations, with maintenance being one of the critical areas of focus. Maintenance is a critical component in organizational settings because it enhances devices' lifespan, extending their usage (Abidi et al., 2022)[1]. However, maintenance needs adequate planning to have intended effects on manufacturing units. Accurate estimation of the equipment failure period massively lowers the risk of breakdowns, accidents, and financial losses (Abidi et al., 2022).[1] For this reason, predictive maintenance has gained popularity in recent decades, being embraced in multiple sectors, including automobiles and aircraft (Abidi et al., 2022)[1]. Machine learning allows manufacturing teams to forecast pending equipment failures. Data analytic tools, like engineering and statistical inferences, assist in predicting the time that the machines are likely to fail, allowing engineers to take appropriate interventions (Abidi et al., 2022)[1]. Ouda et al. (2021)[17] echo the above assertion, postulating that predictive maintenance employs analytical tools to estimate when manufacturing tools need repairs. It monitors equipment health constantly over time, allowing for early detection of failures that could be costly based on historical statistics (Bouabdallaoui et al., 2021)[6]. Therefore, machine learning plays a critical role in manufacturing units by forecasting maintenance.

There are numerous benefits associated with using predictive maintenance to help identify machine failures before they become too problematic or the piece of equipment completely malfunctions. First, Paolanti et al. (2018)[18] report that predictive maintenance is relatively inexpensive and only requires an informal mathematical computation on when, specifically, a piece of equipment requires repair or needs to be replaced. Maintenance can be streamlined and specific to exactly what needs to be repaired or replaced in the most effective and efficient manner possible. Ideally, ML algorithms will predict equipment failure far enough in advance to allow the organization adequate time to address the issue without disrupting production or daily operations (Ren, 2021)[19]. Gustchi et al. (2019)[10] support the above argument, asserting that preventing maintenance techniques trigger repair operations in organized and predefined durations. It allows entities to evaluate equipment maintenance criteria to lower the probability of failures (Gustchi et al., 2019)[10]. Predictive tools utilize time-based and condition-based maintenance to forecast when manufacturing units need checks and repairs. Time-based maintenance continuously monitors equipment's operating time since the last preservation operations (Gutschi et al., 2019)[10]. On the other hand, condition-based maintenance employs specified measurement data about equipment's physical condition, including temperature, vibration, or noise, to determine whether they need repair (Gutschi et al., 2019)[10]. Predictive maintenance techniques apply prognostic models to foretell manufacturing tools' conditions, enabling entities to know the ideal time to undertake repair operations (Gutschi et al., 2019)[10]. Organizations derive reliable forecasting frameworks from the repeated evaluation of data collected from their undertakings. The models are trained to cross-examine and foresee equipment's health, the possibility of failures, or the remaining lifespan (Gutschi et al., 2019)[10]. Cinar et al. (2020)[8] support the above argument, asserting that machine learning is a safe-fire method for maintaining the equipment's safety status through detecting defects and remaining useful life. Cinar et al.

(2020)[8] also posit that the efficacy of predictive maintenance relies heavily on the prevailing and trending information technology (IT) systems and artificial intelligence (AI) technology. Predictive mechanisms rely on several types of data, including sensor information or event-log statistics produced by machine control tools or information technology systems (Gutschi et al., 2019)[10]. Therefore, predictive maintenance tools enable organizations to understand their equipment's conditions and initiate appropriate interventions whenever the need for repairs arises.

ML can be useful for already stressed human resources because it permits workers to focus on other tasks at hand rather than predicting performance maintenance worries. One of the significant benefits of predictive maintenance, per Ayvaz and Alpay (2021)[3], is that it reduces costs associated with human resources and permits the organization's limited human resources to allocate their time elsewhere. The evidence indicates that ML-based predictive maintenance algorithms are time-saving and human-labor-saving tools (Sun et al., 2019)[20]. Betz et al. (2022)[5] support the above outlook, positing that predictive maintenance policy lowers the occurrence of unplanned defects, which inhibit manufacturing teams from undertaking their tasks. Organizations undertake inspection and maintenance actions based on the estimated state of components (Betz et al., 2022)[5]. This approach differs from reactive or corrective maintenance models that are not scheduled, resulting in imbalanced work schedules. Breakdowns occur unexpectedly, making human resources planning difficult. As a result, Betz et al. (2022)[5] argue that reactive maintenance action is more expensive than predictive. Predictive maintenance models streamline resource planning, allowing manufacturing to utilize their human labor efficiently. Malisetty et al. (2017)[14] agree with the above outlook, postulating that many organizations today, including those operating in the IT sector, want to be predictive. They want to gain insights and facts from data analytics to detect patterns and trends that stimulate their growth and anticipate events rather than react to them (Malisetty et al., 2017)[14]. Organizations also rely on the forecast, using what-if simulations to predict what to expect in the future and learn about changes in staff members' behavioral patterns (Malisetty et al., 2017)[14]. The above observations demonstrate that predictive maintenance stemming from machine learning enables entities to maximize their human resources by preventing inefficiencies caused by the equipment breakdown.

To use ML for predictive maintenance purposes, a failure model must be built and implemented. To do so, Sun et al. (2019)[20] explain that enough historical data must be collected to allow algorithms to identify information and predict an equipment failure most of the time. With this data, precise algorithms can be used to isolate the types of failures that might transpire and can be predicted. Moreover, the failure model can identify what the failure process resembles and which precise parts of the equipment are related to the identified failure type. According to Carvalho et al. (2019)[7], the overwhelming majority of machines have a lifespan of years or even decades; therefore, data should be collected over a more extended period to map out the entire system throughout its degradation process. Ideally, data scientists and equipment-specific subject-area experts will collaborate to create a data collection plan. The data has already been collected before a data analyst is brought in to map out the degradation process in many situations. According to Kusumaningrum et al. (2021)[12], predictive maintenance monitors the actual condition of a machine, operating efficiency, and other critical indicators to maximize and prolong intervals between repair operations. This plan minimizes the cost of unplanned downtime or breakdown of manufacturing units (Kusumaningrum et al., 2021)[12]. The method uses several historical data to determine the status of machines, including thermography and tribology. Data acquired using the above observations allows production teams to determine the average life statistics for machines, like the mean time for failure or breakdown, and, in turn, schedule repair activities. Even though the generation of knowledge regarding statistical life information is relatively simple, deriving accurate and detailed mathematic models for forecasting is work-intensive and time-consuming (Gutschi et al., 2019)[10]. In some instances, it could be impossible to require organizations utilizing predictive maintenance to employ data-driven approaches to estimate the probability of their equipment's failure or breakdown (Gutschi et al., 2019)[10]. At the same time, studies are yet to prove the profitability of creating stable data-driven log-based models (Karuppusamy, 2020)[13]. Current literature suggests that predictive maintenance models yield desired outcomes when organizations use the right historical data generation frameworks despite the above concerns.

The first step in using ML for predictive maintenance purposes is to frame the problem that the organization is attempting to address with artificial intelligence (AI). Amruthnath et al. (2018)[2] explain that data experts must determine the type of outputs, whether there exists ample historical and static data, the right strategy to label the recorded events, the portion of events available for each type of failure, and how early in the degradation process the model should be able to predict an equipment failure. Once these and similar pieces of data are identified, subject-area experts in ML and the target equipment can determine collectively which modeling strategy is the best fit (Bampoula et al., 2021)[4]. According to Karuppusamy (2020)[13], the ML algorithm allows manufacturing teams to detect faults in production equipment with the assistance of massive collected datasets. However, organizations struggle to identify the proper ML techniques for their industrial systems. This challenge emanates from the older lean management systems that do not align with today's technological tools. ML combines with predictive maintenance to maintain manufacturing devices through regular interval checks despite the above concerns. The collected data allows predictive maintenance tools to predict the exact time when the equipment will break. This allows companies to schedule repair operations based on historical data (Karuppusamy, 2020)[13]. Karuppusamy (2020)[13] further posits that ML allows companies to precisely schedule many reactive maintenance activities and acquire the needed resources, including spare parts, at the right time. At the same time, manufacturing units can optimize maintenance costs by eliminating expenses for installing sensors and creating physical models (Karuppusamy, 2020)[13].

There are numerous modeling strategies associated with predictive maintenance and optimization. Cheng et al. (2020)[7] stress that regression models are frequently utilized to predict the remaining useful lifetime (RUL) of equipment. Their outputs include specifying how much time is left before the system fails using labeled static and historical datasets. Classification models are a second strategy often employed to predict the likelihood of a piece of equipment failing within a certain timeframe. This strategy uses classification failures to predict the likelihood of a piece of equipment failing within a certain timeframe. The same types of data are used in regression models and classification models. Essentially, regression and classification methods work by modeling the various relationships between equipment features and the degradation trajectory of a system (Susto et al., 2014)[21]. Through-life Engineering (TES) is the other predictive maintenance tool organizations utilize to assess their equipment and determine when they need repairs. According to Okoh and Mehnen (2017)[16] applies advanced technologies to monitor the conditions of machines to assist operators in reducing downtime and product availability. TES is a knowledge-based framework because it combines computational intelligence and experience to create solutions to system challenges. Decision-making relies heavily on the simulation of the prognostic model, which depends on estimated parameters (Jung et al., 2021)[11]. The different outputs help domain experts acquire an ideal sense of judgment in the planning of spare parts acquisition, scheduling repairs, and reducing service downtime caused by breakdowns or machine inefficiencies (Okoh & Mehnen, 2017)[16]. Furthermore, organizations use simulations to determine whether to scrap entire assembly units or replace worn-out parts with brand new ones ((Okoh & Mehnen, 2017)[16].

A third ML-based strategy to predict equipment degradation involves flagging anomalous behavior. Zenisek et al. (2019)[25] note that this strategy is excellent when historical data is limited or failure rates for a piece of equipment are low. The model asks whether specific equipment behavior is normal or abnormal. Static and historical datasets are required, but mostly the labels are unknown because of an inadequate number of data points on system failures. The model assumes that it is possible to precisely define and label normal versus abnormal behavior as these behaviors are related to equipment degradation over time (Traini et al., 2019)[22]. The predictive maintenance model detects anomalous behaviors in machines because it is underpinned by multiple technologies, including the Internet of Things (IoT), Deep Learning (DL), and communication technology (Givnan et al., 2022)[9]. At the same time, the components of the Industry 4.0 revolution allow manufacturing teams to monitor their equipment in real-time. The constant surveillance enables manufacturing teams to assess the health of production devices and identify imminent faults (Givnan et al., 2022)[9]. The above features are essential in equipment since breakdowns due to tear and wear are inevitable.

Other models utilize data-based approaches to predict the possibility of equipment and system failure, enabling organizations to take corrective measures before the unfortunate event occurs. According to Wang et al. (2017)[24], survival models, for example, are based on if the risk of failure changes over time given specific sets of characteristics. Static data is used, as well as information on the specific failure reported timeline. Wang et al. (2017)[24] posit that many types of equipment, including automated teller machines (ATMs) and medical devices, generate system messages that allow those managing them to predict failures. The machines display error events and log files that facilitate the detection of abnormal operations. Tran et al. (2018)[23] echo the above argument, asserting that predictive maintenance tools offer diagnostic insight into the status of machines, enabling management teams to determine the issues to prioritize based on severity. This approach reduces misjudgment, and human errors increase maintenance costs.

In summation, AI and ML have numerous cost-saving benefits, and algorithms can be created for predictive maintenance and optimization purposes. Most organizations have some equipment that must be functional and optimized to perform daily operations. ML-learning tools can provide organizations with early indicators that their equipment is about to fail so that they can take remedial action quickly and prevent shutdowns. ML and predictive maintenance allow firms to detect faults early and develop appropriate solutions. This approach allows manufacturing plants to lower the cost of maintaining their systems and avoid equipment failures that could cause accidents.

CHAPTER 3

3.1 Data overview

Since the oil and gas sector data is not openly available and is scarce, the data sets for this had to be searched online. The reason behind the scarcity of maintenance data is that such information is not made public by the companies. Then the preventive maintenance system is different for each module of the system installed in the oil and gas sector; for example, the data for maintenance of boilers will be managed as standalone same goes with the scrubbers etc. the online data sets that have been researched is based on the preventive data based on the machine errors which occur during routine and turnaround procedures, and in these datasets, the process periods and what the error type was also mentioned, as a matter of discussion the data type of the data sets are structured which is based on the data points meantime of the preventive maintained and their occurrences are mentioned. There are 10,000 instances or data points stored in rows with 14 features in columns. The data points have been labelled as L, M and H, which are short for Low, Medium. The data sets have machine failure, which consists of five independent failure modes, which are as follows

Failure Mode	Preconditions	Instances in datasets	Description
Tool Wear Failure (TWF)	Temperature 200 degrees for 240 mins	120 Times (Tool Replaced: 69, Fails 51 Times)	In this mode the tool will be replaced of fail at a randomly selected time
Heat Dissipation Failure (HDF)	If the temperature difference between air and process is 8.6 K	115 Datapoints	Heat dissipation causes process failures
Power Failure (PWF)	If the power is 3500 W or above 9000 W	95 datapoints	The product of torque and rotational speed (in rad/s)

			equals the power required for the process.
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)	98 data points	The process fails due to overstrain
Random Failure	NO preconditions	5 data points	Each process has a chance of 0,1 % to fail regardless of its process parameters.

CHAPTER 4

4.1 Data Analysis

4.1.1 DATA UNDERSTANDING:

Understanding of dataset is very important in any project and its an essential step on the basis of which the appropriate algorithms are chosen and applied to solve the problem. In this project we have industrial dataset which aims to solve the problem of detecting the machine failure and we have different readings of the properties of the machines which can help to determine the failure and the change of a failure in order to prepare or prevent it before it happens.

Following are the most important features in our dataset:

- air temperature
- process temperature
- rotational speed
- torque
- tool wear

We also have different types of failure types but since the dataset classes are found to be highly imbalanced and we have very few cases of failure so we chose to ignore the specific types of failures and only use the failure column for this problem which will be 1 for all types of failures.

- Libraries Used:

```
library(ggplot2)
library(dplyr)
library(caTools)
library(caret)
library(randomForest)
library(ROSE)
library(e1071)
```

4.1.2 DATA EXPLORATION:

- Loading the dataset
- Looking at the structure of dataset

```
> data <- read.csv("ai4i2020.csv")
> str(data)
'data.frame':	10000 obs. of  14 variables:
 $ UDI                    : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Product.ID             : chr  "M14860" "L47181" "L47182" "L47183" ...
 $ Type                   : chr  "M" "L" "L" "L" ...
 $ Air.temperature..K.    : num  298 298 298 298 298 ...
 $ Process.temperature..K.: num  309 309 308 309 309 ...
 $ Rotational.speed..rpm. : int  1551 1408 1498 1433 1408 1425 1558 1527 1667 1741 .
 $ Torque..Nm.            : num  42.8 46.3 49.4 39.5 40 41.9 42.4 40.2 28.6 28 ...
 $ Tool.wear..min.        : int  0 3 5 7 9 11 14 16 18 21 ...
 $ Machine.failure        : int  0 0 0 0 0 0 0 0 0 0 ...
 $ TWF                    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ HDF                    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ PWF                    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ OSF                    : int  0 0 0 0 0 0 0 0 0 0 ...
 $ RNF                    : int  0 0 0 0 0 0 0 0 0 0 ...
```

- Dataset summary

```
> summary(data)
      UDI          Product.ID            Type           Air.temperature..K.
 Min.   :    1   Length:10000       Length:10000       Min.   :295.3
 1st Qu.: 2501   Class :character   Class :character   1st Qu.:298.3
 Median : 5000   Mode  :character   Mode  :character   Median :300.1
 Mean   : 5000                                         Mean   :300.0
 3rd Qu.: 7500                                         3rd Qu.:301.5
 Max.   :10000                                         Max.   :304.5
 Process.temperature..K. Rotational.speed..rpm.  Torque..Nm.     Tool.wear..min.
 Min.   :305.7           Min.   :1168           Min.   : 3.80   Min.   :  0
 1st Qu.:308.8           1st Qu.:1423           1st Qu.:33.20   1st Qu.: 53
 Median :310.1           Median :1503           Median :40.10   Median :108
 Mean   :310.0           Mean   :1539           Mean   :39.99   Mean   :108
 3rd Qu.:311.1           3rd Qu.:1612           3rd Qu.:46.80   3rd Qu.:162
 Max.   :313.8           Max.   :2886           Max.   :76.60   Max.   :253
 Machine.failure        TWF               HDF               PWF               OSF
 Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000   Min.   :0.0000
 1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.0000
 Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000   Median :0.0000
 Mean   :0.0339   Mean   :0.0046   Mean   :0.0115   Mean   :0.0095   Mean   :0.0098
 3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.0000
 Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000   Max.   :1.0000
      RNF
 Min.   :0.0000
 1st Qu.:0.0000
 Median :0.0000
 Mean   :0.0019
 3rd Qu.:0.0000
 Max.   :1.0000
```

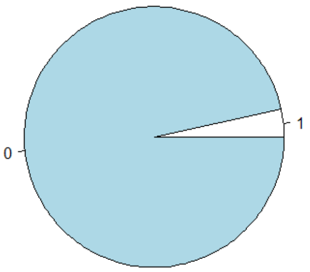
- Looking for Missing Values

```
> colSums(is.na(data))
                    UDI              Product.ID                    Type
                      0                       0                       0
    Air.temperature..K. Process.temperature..K.  Rotational.speed..rpm.
                      0                       0                       0
            Torque..Nm.         Tool.wear..min.         Machine.failure
                      0                       0                       0
                    TWF                     HDF                     PWF
                      0                       0                       0
                    OSF                     RNF
                      0                       0
```

Luckily, we have found no missing values in our dataset so there is no data cleaning required and all the columns already have the float values and there is no missing or any other datatype than float.

- Finding total number of machine failures in our dataset

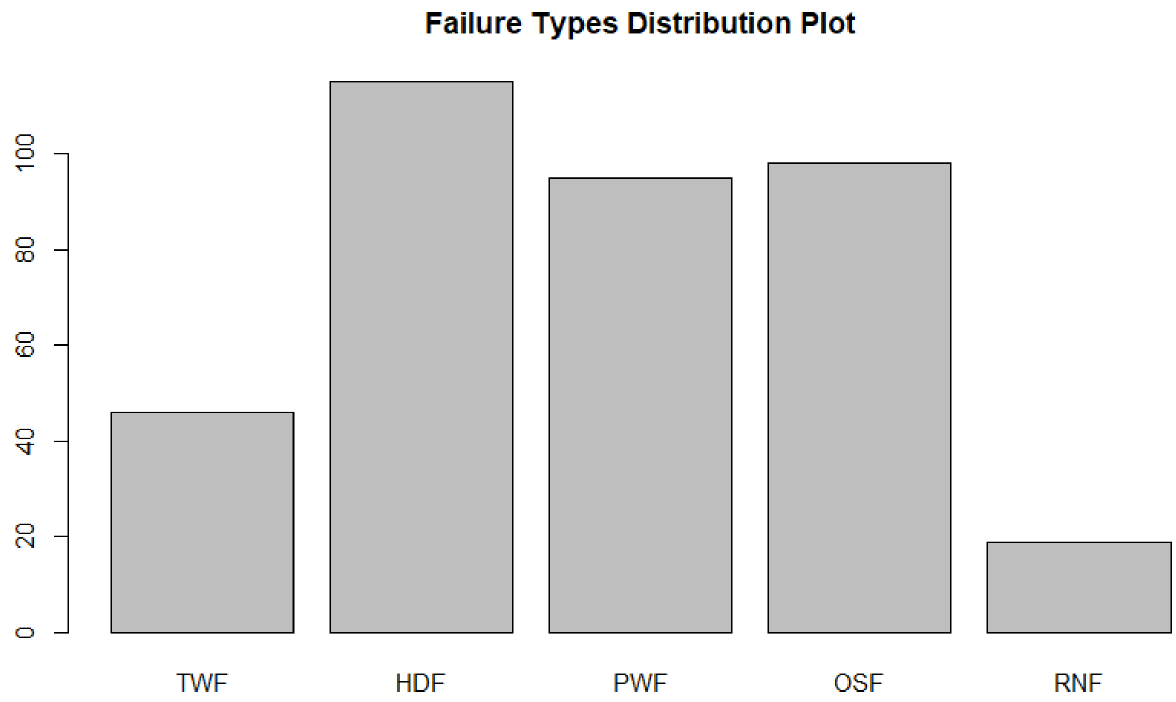
```
> v1 <- data %>%
+   group_by(Machine.failure) %>% # Variable to be transformed
+   count()%>%
+   ungroup() %>%
+   mutate(perc = `n` / sum(`n`)) %>%
+   arrange(perc)
> v1
# A tibble: 2 × 3
  Machine.failure     n   perc
            <int> <int>  <dbl>
1               1   339 0.0339
2               0  9661 0.966
>
```



After visualizing the classes, we came to know that we have a dataset which is highly imbalanced, and we should have an approach which should work even if the dataset is imbalanced. As in most cases the classes are balanced and the traditional evaluation metrics like accuracy work in those cases but not in the cases where we have highly imbalanced classes.

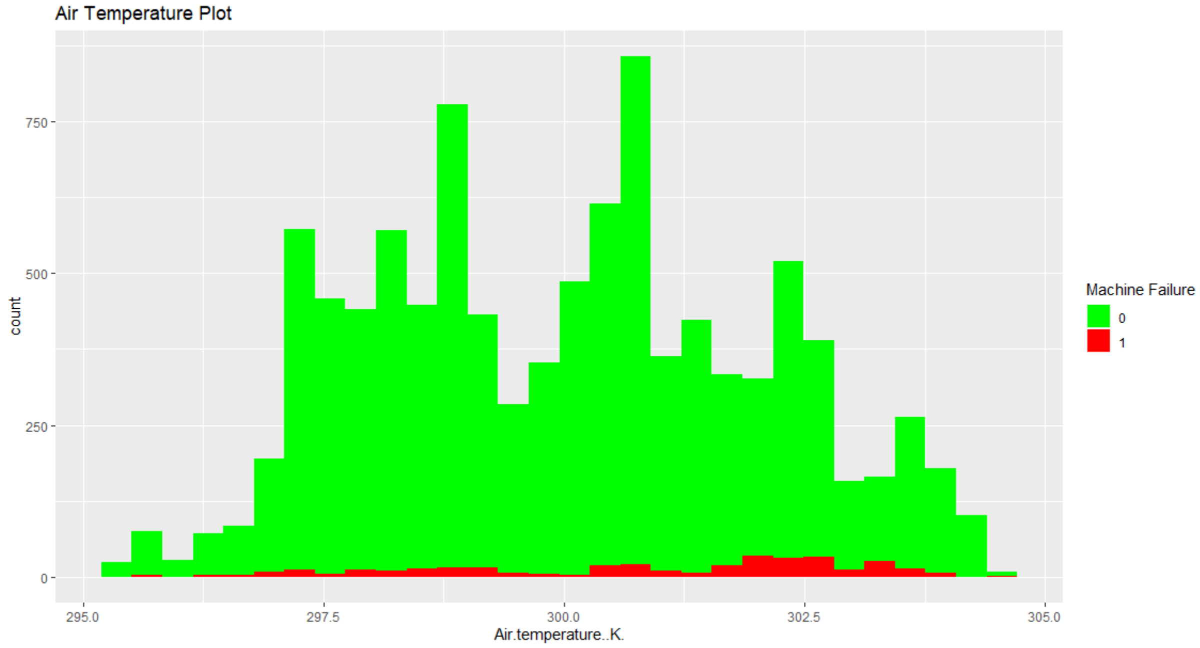
We decided to have one failure class which will take care of all the different failure types because we already have imbalanced dataset where very few classes are of failure and further classifying them in different types will make it even harder for the algorithm to predict.

- Different Failure Types Distribution



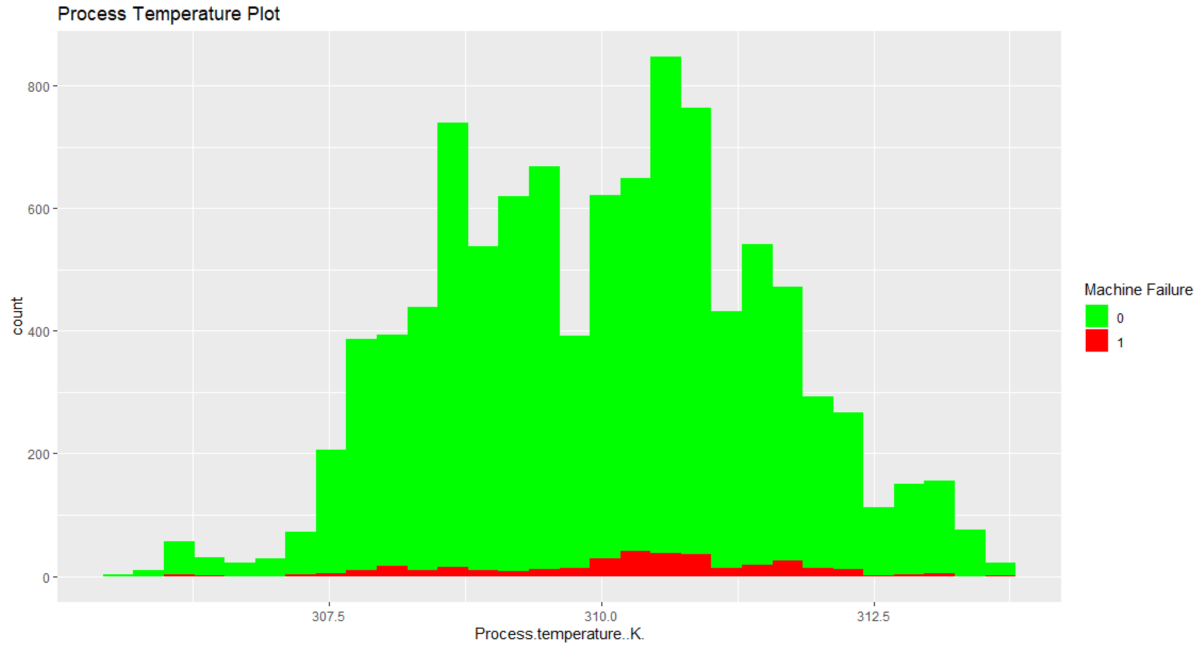
This plot provides us the information of different types of failures and which one of them occurred more frequently and HDF was the one which occurred more frequently. Although we have decided to consider all failures as a single failure class.

- Air Temperature Plot



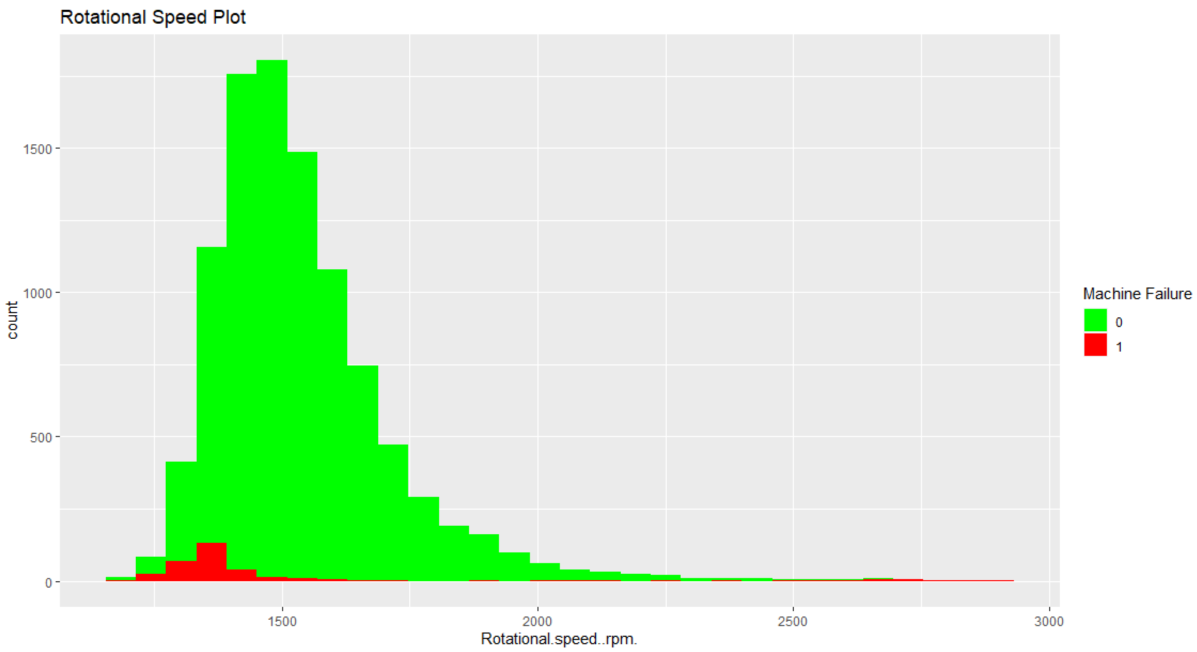
This plot shows the failures occurred more around air temperature value close to 300 but there are a lot of values where the temperature is around 300 but they weren't failures.

- Process Temperature Plot

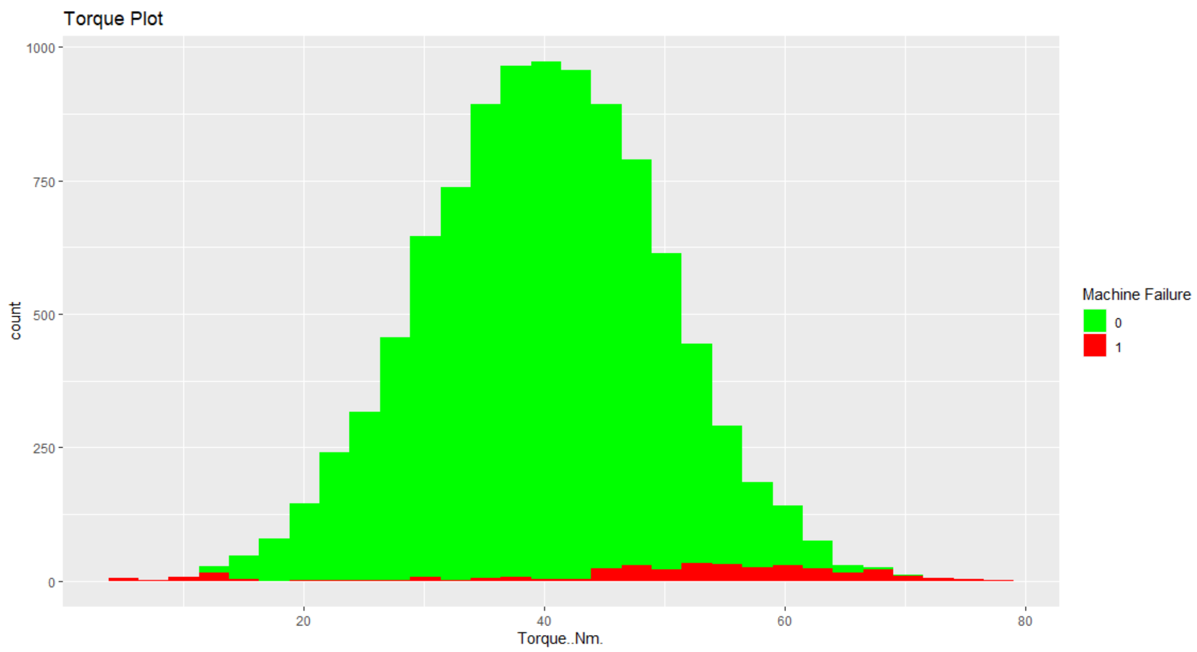


This plot shows the pressure temperature is mostly around 310 to 311 for both type of classes weather its failure or non failure.

- Rotational Speed Plot



- Torque Plot



This is an interesting plot as compared to the rest because of the fact that most failures are after 45 torque, but the rest of the non-failure majority cases are around 40 and spread evenly on both sides but the failure cases occurred more after 45 torque. Which provides us the insight that this feature will be very important in the classification.

- Tool Wear Plot

This plot tells us although failure cases are spread across different values of tool wear but most of them are found at 220 value.

4.1.3 Overall Observation:

The very first observation after looking at the graphs is that we have a problem which has highly imbalanced dataset. We have 9661 data instances where we have no failure and only 339 data instances where we have a failure. The highly imbalanced dataset requires us to choose the appropriate learning algorithm for this problem and choose the appropriate evaluation metrics as using accuracy evaluation metrics can be very misleading in the case of highly imbalanced dataset.

From the above plots, it can be seen that at some points the frequency of the failure cases increases as the red bars are high at certain points in the distribution plot. Which is a good indicator that these features can be valuable to learn a pattern when the machine will fail and can be helpful in finding the failure.

4.2 Data Pre-Processing:

- Selecting the necessary columns
- Changing the datatype

```
df <- select(data,c("Type","Air.temperature..K.","Process.temperature..K.",
                    "Rotational.speed..rpm.","Torque..Nm.","Tool.wear..min.",
                    "Machine.failure"))

df$Type <- as.factor(df$Type)

str(df)
```

```
> str(df)
'data.frame':   10000 obs. of  7 variables:
 $ Type                   : Factor w/ 3 levels "H","L","M": 3 2 2 2 2 3 2 2 3 3 ...
 $ Air.temperature..K.    : num  298 298 298 298 298 ...
 $ Process.temperature..K.: num  309 309 308 309 309 ...
 $ Rotational.speed..rpm. : int  1551 1408 1498 1433 1408 1425 1558 1527 1667 1741 ...
 $ Torque..Nm.            : num  42.8 46.3 49.4 39.5 40 41.9 42.4 40.2 28.6 28 ...
 $ Tool.wear..min.        : int  0 3 5 7 9 11 14 16 18 21 ...
 $ Machine.failure        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
```

First step we did in pre-processing is to select only the necessary columns as the ID's columns are not going to be helpful in classification. After filtering the columns we had to make sure that we have the right datatypes so we convert the Type column to factor.

- Scaling the data

```
# Scaling
data_pre <- preProcess(df, method="range")
df <- predict(data_pre, df)

head(df)
```

Second step we did is to scale the data as it will be very helpful for the machine learning algorithms to learn when the data is scaled.

- Scaling the data

```
> head(df)
  Type Air.temperature..K. Process.temperature..K. Rotational.speed..rpm. Torque..Nm. Tool.wear..min.
1    M           0.3043478               0.3580247              0.2229336   0.5357143      0.00000000
2    L           0.3152174               0.3703704              0.1396973   0.5837912      0.01185771
3    L           0.3043478               0.3456790              0.1920838   0.6263736      0.01976285
4    L           0.3152174               0.3580247              0.1542491   0.4903846      0.02766798
5    L           0.3152174               0.3703704              0.1396973   0.4972527      0.03557312
6    M           0.3043478               0.3580247              0.1495925   0.5233516      0.04347826
  Machine.failure
1               0
2               0
3               0
4               0
5               0
6               0
```

- Training and Testing Data

```
# Making training and testing data
set.seed(100)

split = sample.split(df$Machine.failure, SplitRatio=0.8)
train = subset(df, split == TRUE)
test = subset(df, split == FALSE)

nrow(train)
nrow(test)
```

We split the data by 80% so we have 8000 rows for training the model and 2000 rows on which the model will be evaluated.

4.3 Data Modeling:

In data modelling we will use the training dataset to train the model and evaluate the model performance on the unseen testing dataset.

As our dataset has a very imbalanced classes so we will choose the algorithms which are known to perform better in such case scenarios. Also, in the case of highly imbalanced classes the evaluation metrics is of very importance as just looking at the model accuracy doesn’t provide the full information about the model performance. That’s why for the model evaluation we will look at the confusion metrics and ROC curve and evaluate the models based on these metrics.

4.3.1 LOGISTIC REGRESSION:

Logistic regression is a good classification algorithm which is mostly considered as a baseline model before applying other complex models to classify the dataset. It's a good choice when data is linearly separable and its easier to implement and its very efficient.

- Logistic regression

```
# Logistic Regression
log_model <- glm(Machine.failure ~.,family=binomial(link='logit'),data=train)

summary(log_model)

res_test <- predict(log_model,test)
res_test_lr <- ifelse(res_test > 0.5,1,0)
error_test <- mean(res_test_lr != test$Machine.failure)
print(paste('Accuracy',1-error_test))
```

The accuracy we got after applying Logistic Regression model is 96.8%

- Training Dataset Confusion Metrics:

```
res_train <- predict(log_model,train)
res_train_lr <- ifelse(res_train > 0.5,1,0)
error_train <- mean(res_train_lr != train$Machine.failure)
print(paste('Accuracy',1-error_train))

cm_train <- confusionMatrix(data = as.factor(res_train_lr), reference = train$Machine.failure,
                            mode = "everything")
cm_train
```

- Training Dataset Confusion Metrics:

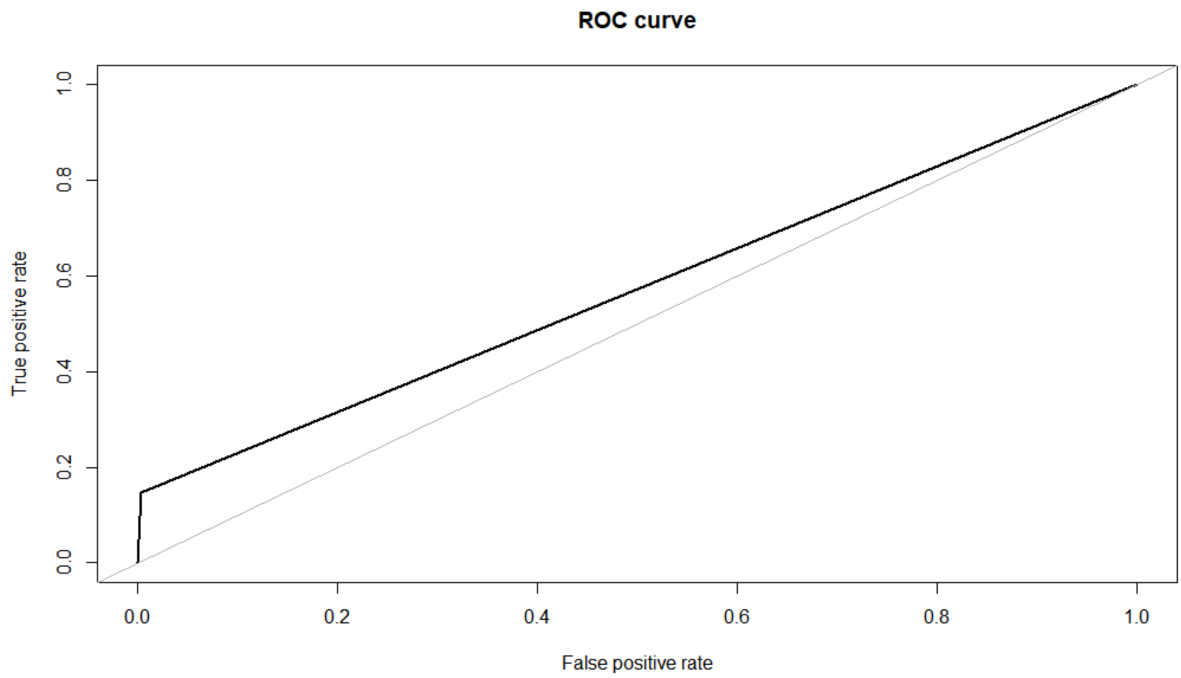
```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1926   58
         1    6   10

               Accuracy : 0.968
                 95% CI : (0.9593, 0.9753)
    No Information Rate : 0.966
    P-Value [Acc > NIR] : 0.3387

                  Kappa : 0.2281

 Mcnemar's Test P-Value : 1.83e-10

            Sensitivity : 0.9969
            Specificity : 0.1471
         Pos Pred Value : 0.9708
         Neg Pred Value : 0.6250
              Precision : 0.9708
                 Recall : 0.9969
                     F1 : 0.9837
             Prevalence : 0.9660
         Detection Rate : 0.9630
   Detection Prevalence : 0.9920
      Balanced Accuracy : 0.5720
```

- Testing Dataset Confusion Metrics:

```
cm_test <- confusionMatrix(data = as.factor(res_test_lr), reference = test$Machine.failure,
                           mode = "everything")
cm_test
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1926   58
         1    6   10

               Accuracy : 0.968
                 95% CI : (0.9593, 0.9753)
    No Information Rate : 0.966
    P-Value [Acc > NIR] : 0.3387

                  Kappa : 0.2281

 Mcnemar's Test P-Value : 1.83e-10

            Sensitivity : 0.9969
            Specificity : 0.1471
         Pos Pred Value : 0.9708
         Neg Pred Value : 0.6250
              Precision : 0.9708
                 Recall : 0.9969
                     F1 : 0.9837
             Prevalence : 0.9660
         Detection Rate : 0.9630
   Detection Prevalence : 0.9920
      Balanced Accuracy : 0.5720
```

- ROC Curve for Logistic Regression:



4.3.2 Support Vector Machine:

SVM is preferred because it uses less computational power and can provide good results in very high dimensional data. Although we don't have a high dimensional data but still we chose to apply it to find out if we can get any better results and if the data in linearly separable.

- SVM

```
# SVM
model_svm = svm(formula = Machine.failure ~ .,
                 data = train,
                 type = 'C-classification',
                 kernel = 'linear')

res_train_svm = predict(model_svm, newdata = train)
res_test_svm = predict(model_svm, newdata = test)
```

- Confusion Metrics on Testing Data:

```
confusionMatrix(data = res_test_svm, reference = test$Machine.failure,
                mode = "everything")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1932   68
         1    0    0

               Accuracy : 0.966
                 95% CI : (0.9571, 0.9735)
    No Information Rate : 0.966
    P-Value [Acc > NIR] : 0.5322

                  Kappa : 0

 Mcnemar's Test P-Value : 4.476e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.9660
         Neg Pred Value :    NaN
              Precision : 0.9660
                 Recall : 1.0000
                     F1 : 0.9827
             Prevalence : 0.9660
         Detection Rate : 0.9660
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000
```

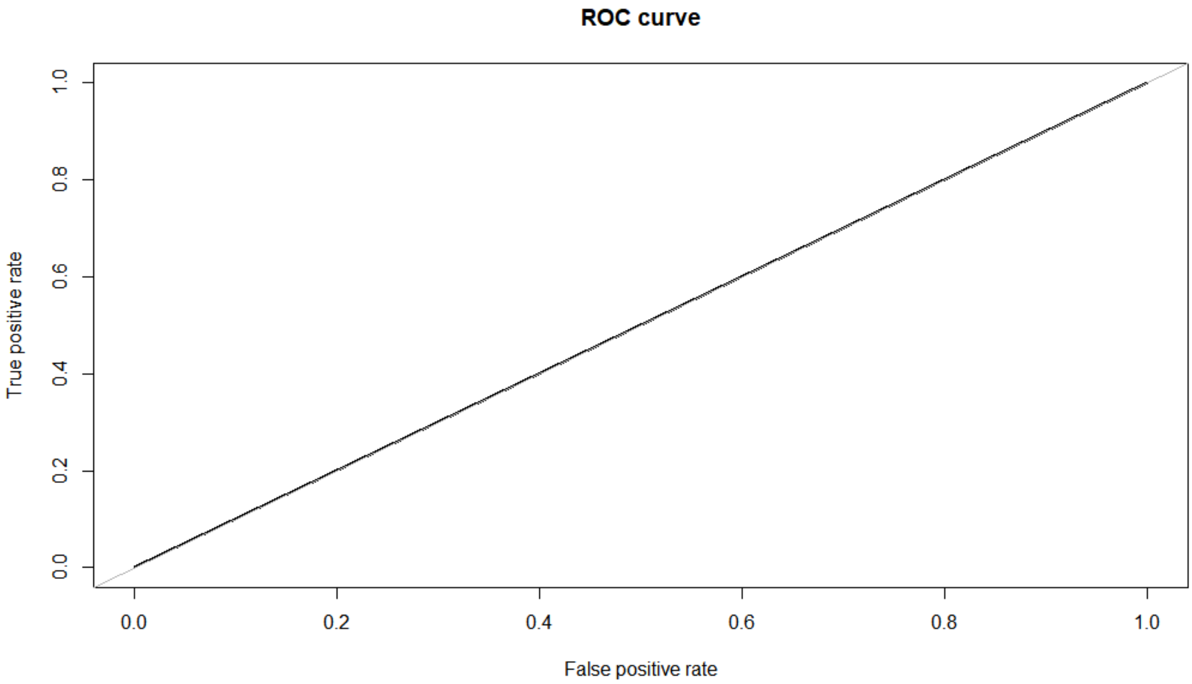
- Confusion Metrics on Training Data:

```
confusionMatrix(data = res_train_svm, reference = train$Machine.failure,
                mode = "everything")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7729  271
         1    0    0

               Accuracy : 0.9661
                 95% CI : (0.9619, 0.97)
    No Information Rate : 0.9661
    P-Value [Acc > NIR] : 0.5162

                  Kappa : 0

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.0000
            Specificity : 0.0000
         Pos Pred Value : 0.9661
         Neg Pred Value :    NaN
              Precision : 0.9661
                 Recall : 1.0000
                     F1 : 0.9828
             Prevalence : 0.9661
         Detection Rate : 0.9661
   Detection Prevalence : 1.0000
      Balanced Accuracy : 0.5000
```

- ROC Curve for SVM:

```
roc.curve(test$Machine.failure, res_test_svm)
```



4.3.3 NAIVE BAYES:

Naïve bayes is another good classification algorithm where the features are independent of one another and can provide better results than other algorithms. Naïve bayes is a probabilistic machine learning algorithm which is a different approach to what we did previously so that is why we chose this algorithm next to apply to this problem.

- Naïve bayes

```
model_nb <- naiveBayes(Machine.failure ~ ., data = train)

res_train_nb = predict(model_nb, newdata = train)
res_test_nb = predict(model_nb, newdata = test)
```

- Confusion Metrics on Training Data:

```
confusionMatrix(data = res_train_nb, reference = train$Machine.failure,
                mode = "everything")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7622  216
         1  107   55

               Accuracy : 0.9596
                 95% CI : (0.9551, 0.9638)
    No Information Rate : 0.9661
    P-Value [Acc > NIR] : 0.9992

                  Kappa : 0.2346

 Mcnemar's Test P-Value : 1.863e-09

            Sensitivity : 0.9862
            Specificity : 0.2030
         Pos Pred Value : 0.9724
         Neg Pred Value : 0.3395
              Precision : 0.9724
                 Recall : 0.9862
                     F1 : 0.9793
             Prevalence : 0.9661
         Detection Rate : 0.9527
   Detection Prevalence : 0.9798
      Balanced Accuracy : 0.5946
```

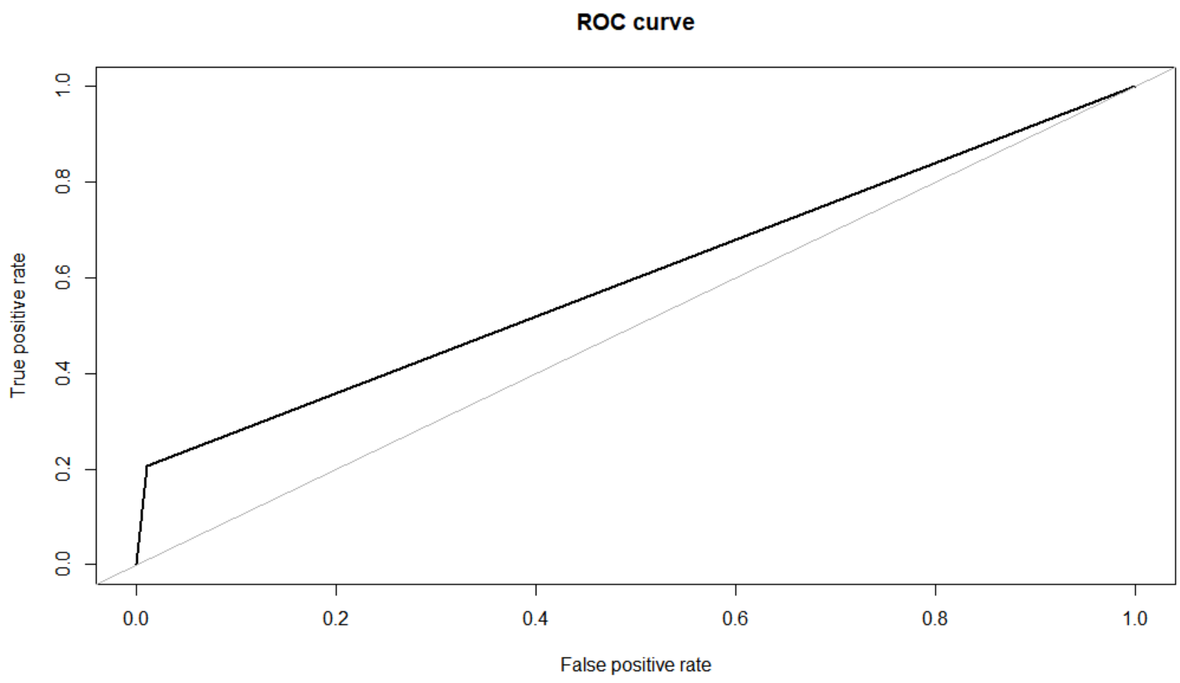
- Confusion Metrics on Test Data:

```
confusionMatrix(data = res_test_nb, reference = test$Machine.failure,
                mode = "everything")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1912   54
         1   20   14

               Accuracy : 0.963
                 95% CI : (0.9538, 0.9708)
    No Information Rate : 0.966
    P-Value [Acc > NIR] : 0.790747

                  Kappa : 0.2577

 Mcnemar's Test P-Value : 0.000125

            Sensitivity : 0.9896
            Specificity : 0.2059
         Pos Pred Value : 0.9725
         Neg Pred Value : 0.4118
              Precision : 0.9725
                 Recall : 0.9896
                     F1 : 0.9810
             Prevalence : 0.9660
         Detection Rate : 0.9560
   Detection Prevalence : 0.9830
      Balanced Accuracy : 0.5978
```

- ROC Curve for Naïve Bayes:

```
roc.curve(test$Machine.failure, res_test_nb)
```



4.3.4 RANDOM FOREST:

Random Forest algorithm which is constructed by decision trees algorithms. This algorithm uses many decision trees to make an ensemble and the final prediction is made by combining the outcome of all the trees in the ensemble.

This algorithm has a speciality of fitting to the complex data as it utilizes the power of decision trees and along with that it tries to not overfit the data which is the common issue of decision trees and for that reason it performs better than decision tree. Because of its power of fitting to the complex data we are using this algorithm to apply to our dataset.

- Random Forest

```
# Random Forest
model_RF = randomForest(x = train,y = train$Machine.failure,
                        ntree = 500)

res_train_rf = predict(model_RF, newdata = train)
res_test_rf = predict(model_RF, newdata = test)
```

- Confusion Metrics on Training Data:

```
confusionMatrix(data = res_train_rf, reference = train$Machine.failure,
                mode = "everything")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 7729    0
         1    0  271

               Accuracy : 1
                 95% CI : (0.9995, 1)
    No Information Rate : 0.9661
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.0000
            Specificity : 1.0000
         Pos Pred Value : 1.0000
         Neg Pred Value : 1.0000
              Precision : 1.0000
                 Recall : 1.0000
                     F1 : 1.0000
             Prevalence : 0.9661
         Detection Rate : 0.9661
   Detection Prevalence : 0.9661
      Balanced Accuracy : 1.0000
```

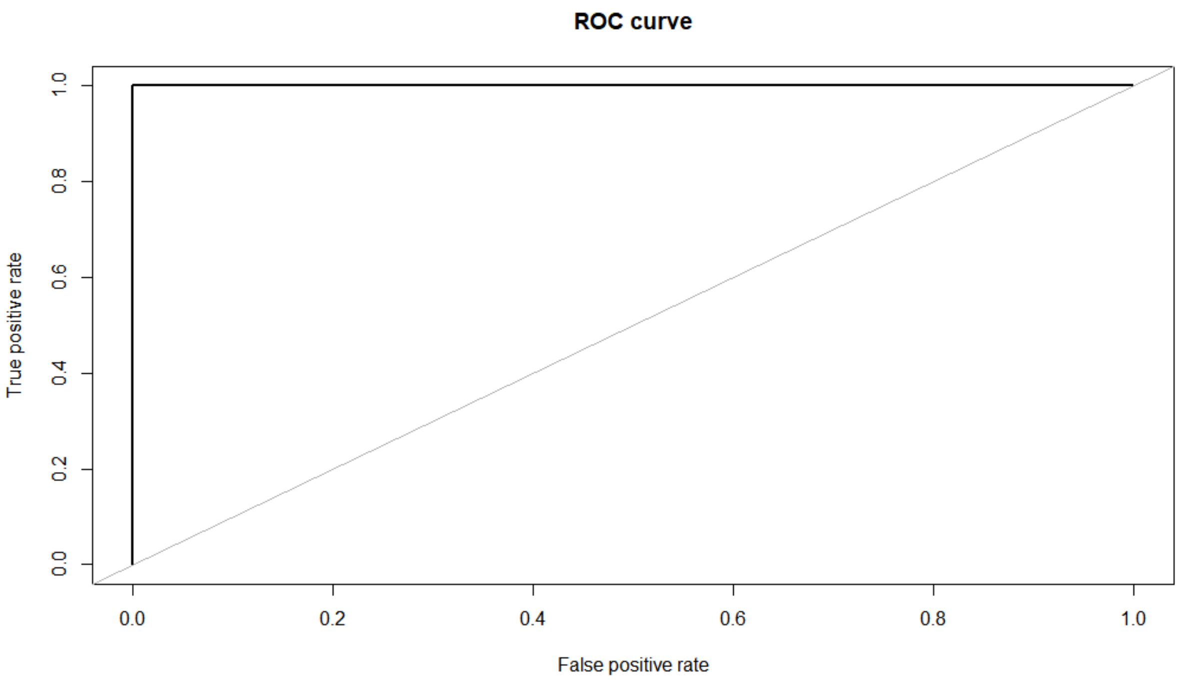
- Confusion Metrics on Testing Data:

```
confusionMatrix(data = res_test_rf, reference = test$Machine.failure,
                mode = "everything")
```

```
Confusion Matrix and Statistics

          Reference
Prediction    0    1
         0 1932    0
         1    0   68

               Accuracy : 1
                 95% CI : (0.9982, 1)
    No Information Rate : 0.966
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1

 Mcnemar's Test P-Value : NA

            Sensitivity : 1.000
            Specificity : 1.000
         Pos Pred Value : 1.000
         Neg Pred Value : 1.000
              Precision : 1.000
                 Recall : 1.000
                     F1 : 1.000
             Prevalence : 0.966
         Detection Rate : 0.966
   Detection Prevalence : 0.966
      Balanced Accuracy : 1.000
```

- ROC Curve for Random Forest:

```
roc.curve(test$Machine.failure, res_test_rf)
```



CHAPTER 5

5.1 Results:

Model	Sensitivity	Specificity	Precision	Recall	F1	Area Under Curve (AUC)
Logistic Regression	99	14	97	99	98	57.2
Support Vector Machine	100	0	96	100	98	50.0
Naïve Bayes	98	20	97	98	98	59.8
Random Forest	100	100	100	100	100	100

From the results it can be seen that Random Forest algorithm outperformed all the other machine learning algorithms and not only it performed well but gave us 100% results on both the training and testing datasets. Random Forest fit perfectly on our dataset and our assumption of thinking that it will perform better because of its tendency to fit to the very complex data was proven correct after looking at the results.

This model is capable of identifying the machine failure before it actually happens by looking at the features provided to us prediction. These results can be crucial in maintaining the machinery and avoiding any failures by taking proper measures for the machines which are near to failure.

5.2 Conclusion:

In this project we are required to build a model which should be able to predict the failure of a machine to avoid the future machinery failures. For this purpose, we did analysis of data and chose machine learning algorithms which we thought would perform well on the data as our data contained very imbalanced classes. Upon applying different linear models, we came to know that the classes are not linearly separable as the linear models performed very bad in classifying the failure of the machine.

So, after knowing that we are working with complex data we applied other algorithms with more learning power. We also scaled our data as a pre-processing step and decided to consider the evaluation metrics to be the sensitivity, specificity, precision, recall and F1 scores because of imbalanced classes because just looking at accuracy wouldn't provide sufficient information about the model's performance. After all these steps we found Random Forest algorithm to be the best performing one and choose that as our final model.

5.2.1 Recommendation and Future Work

To adopt the approach presented in this project, the oil and gas companies can choose an integrated plan that expands on the array and quality of data so that testing of the algorithms for predicting maintenance that applies to their system without using manual processes involving employees. In effect, prediction using AI tools will reduce the staffing needs by precisely pinpointing which equipment and assemblies need maintenance which in turn will enhance the decision-making process to ensure that the minimum workforce is utilized for high-profit maximization because the model arrived at in this project has a positive cost-benefit analysis. Adopting the model recommended in this project will enable the company to generate additional features for simulation which improves data preprocessing to a greater extent. This will be possible if the model is integrated with superior analytical models (such as reservoir models) that are cloud-based for high equipment performance capabilities.

The model contains complex but essential parameters believed to have the greatest impact on operational and asset equipment concerning uptime because risks are failures that are dealt with in advance before they occur. This will reduce the cost of the maintenance and schedule and predict maintenance before time, increasing the availability of the system, and thus increasing the throughput, which will eventually increase the revenues. The model is a perfect way of designing and deploying a system that will transfer the scheduled preventive maintenance into a coordinated predictive maintenance system. by providing the proper training of the algorithm and constant feedback and upgrades because the proposed system has been fully tested using the current predictive software to detect any failure before it happens and adequately plan out the maintenance activity.

5.2.2 Future Work

Predictive analytics and maintenance models are now the most extreme examples of scaled, augmented decision-making. Future models should allow all of the aforementioned maintenance analytics and the corresponding predictive, planned preventative, condition-based, descriptive, and dynamic strategies to be combined by operations and asset management teams to maximize maintenance and operational costs, asset lifecycle management, current productivity, and market demand.

REFERENCES

1- Abidi, M.H., Mohammed, M.K. & Alkhalefah, H. (2022). Predictive maintenance planning for industry 4.0 using machine learning for sustainable manufacturing. *Sustainability*, *14*(3387). https://doi.org/10.3390/su14063387.

2- Amruthnath, N., & Gupta, T. (2018, April). A research study on unsupervised machine learning algorithms for early fault detection in predictive maintenance. In *2018 5th international conference on industrial engineering and applications (ICIEA)* (pp. 355-361).

3- Ayvaz, S., & Alpay, K. (2021). Predictive maintenance system for production lines in manufacturing: A machine learning approach using IoT data in real-time. *Expert Systems with Applications*, *173*, 114598. https://doi.org/10.1016/j.eswa.2021.114598.

4- Bampoula, X., Siaterlis, G., Nikolakis, N., & Alexopoulos, K. (2021). A deep learning model for predictive maintenance in cyber-physical production systems using lstm autoencoders. *Sensors (Basel, Switzerland)*, *21*(3). https://doi.org/10.3390/s21030972.

5- Betz, W., Papaioannou, I., Zeh, T. et al. (2022). Data-driven predictive maintenance for gas distribution networks. *ASCE-ASME J. Risk Uncertainty Eng. Syst., Part A: Civ. Eng.*, *8*(2), 1-10.

6- Bouabdallaoui, Y., Lafhaj, Z., Yim, P., Ducoulombier, L., & Bennadji, B. (2021). Predictive maintenance in building facilities: A machine learning-based approach. *Sensors (Basel, Switzerland)*, *21*(4), 1044. https://doi.org/10.3390/s21041044.

7- Cheng, J. C., Chen, W., Chen, K., & Wang, Q. (2020). Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms. *Automation in Construction*, *112*, 103087. https://doi.org/10.1016/j.autcon.2020.103087.

8- Cinar, Z.M., Nuhu, A.A., Zeehsna, Q. et al. (2020). Machine learning in predictive maintenance towards sustainable smart manufacturing in industry 4.0. *Sustainability, 12*(8211). doi:10.3390/su12198211.

9- Givnan, S., Chalmers, C., Fergus, P. Ortega-Martorell, S. & Whalley, T. (2022). Anomaly detection using autoencoder reconstruction upon industrial motors. *Sensors, 22*(3166). https://doi.org/10.3390/s22093166.

10- Gutschi, C., Furian, N., Suschnigg, J. et al. (2019). Log-based predictive maintenance in discrete parts manufacturing. *Procedia CIRP, 79*, 528-533.

11- Jung, K., Kashyap, S., Avati, A., et al. (2021). A framework for making predictive models useful in practice. *Journal of the American Medical Informatics Association: JAMIA*, *28*(6), 1149–1158. https://doi.org/10.1093/jamia/ocaa318.

12- Kusumaningrum, D., Kurniati, N. & Santosa, B. (2021). Machine learning for predictive maintenance. *Proceedings of the International Conference on Industrial Engineering and Operations Management Sao Paulo, Brazil*, 2348-2356.

13- Karuppusamy, P. (2020). Machine learning approach to predictive maintenance in manufacturing industry- a comparative study. *Journal of Soft Computing Paradigm (JSCP), 2*(4), 246-255.

14- Malisetty, S., Archana, R.V. & Kumari, V. (2017). Predictive analytics in HR management. *Indian Journal of Public Health Research and Development, 8*(3). DOI:10.5958/0976-5506.2017.00171.1.

15- Montero Jimenez, J., Schwartz, S., Vingerhoeds, R., Grabot, B., & Salaün, M. (2020). Towards multi-model approaches to predictive maintenance: A systematic literature survey on diagnostics and prognostics. *Journal of Manufacturing Systems*, *56*, 539-557.

16- Okoh, C. & Mehnen, R.J. (2017). Predictive maintenance modelling for through-life engineering services. *Procedia CIRP*, *59*, 196-201.

17- Ouda, E., Maalouf, M. & Sleptchenko, A. (2021). Machine learning and optimization for predictive maintenance based on predicting failure in the next five days. In *Proceedings of the 10th International Conference on Operations Research and Enterprise Systems (ICORES 2021)*, 192-199. DOI: 10.5220/0010247401920199.

18- Paolanti, M., Romeo, L., Felicetti, A., Mancini, A., Frontoni, E., & Loncarski, J. (2018, July). Machine learning approach for predictive maintenance in industry 4.0. In *2018 14th IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA)* (pp. 1-6). IEEE. DOI: 10.1109/MESA.2018.8449150

19- Ren, Y. (2021). Optimizing predictive maintenance with machine learning for reliability improvement. *ASCE-ASME J Risk and Uncert in Engrg Sys Part B Mech Engrg*, *7*(3).

20- Sun, Y., Xu, Z., & Zhang, T. (2019). *On-board predictive maintenance with machine learning* (No. 2019-01-1048). SAE Technical Paper. DOI:

21- Susto, G. A., Schirru, A., Pampuri, S., McLoone, S., & Beghi, A. (2014). Machine learning for predictive maintenance: A multiple classifier approach. *IEEE Transactions on Industrial Informatics*, *11*(3), 812-820. DOI: 10.1109/TII.2014.2349359

22- Traini, E., Bruno, G., D’antonio, G., & Lombardi, F. (2019). Machine learning framework for predictive maintenance in milling. *IFAC-PapersOnLine*, *52*(13), 177-182.

23- Tran, D., Dabrowski, K. & Skrzypek, K. (2018). The predictive maintenance concept in the maintenance department of the "industry 4.0" production enterprise. *Foundations of Management, 10*(1): 283-292. DOI:10.2478/fman-2018-0022.

24- Wang, J., Li, C., Han, S., Sarkar, S., & Zhou, X. (2017). Predictive maintenance based on eventlog analysis: A case study. *IBM Journal of Research and Development*, *61*(1), 11-121. DOI: 10.1147/JRD.2017.2648298

25- Zenisek, J., Holzinger, F., & Affenzeller, M. (2019). Machine learning based concept drift detection for predictive maintenance. *Computers & Industrial Engineering*, *137*, 106031.