Individual Machine Learning Project

Russell Matthew Bayes

28/03/2022

1. INTRODUCTION

This Rmarkdown is part of the final assignment of the Professional Data Science Certificate program, run by HarvardX on the EDx platform.

This assignment demonstrates the various statistical methods and Machine Learning techniques that have been taught, through the modules of the professional certificate series, to an independent data set. In meeting the assigned project requirements, a topic has been selected that is both data-rich and has few publicly available write ups.

The **Machine Failure Prediction Data set** from the Kaggle platform was chosen several reasons: it can be accessed and understood by a wide audience, and it facilitates the development of advanced data visualization techniques, and machine learning models.

This report analysis in detail the Machine Failure data set in five steps:

1. **Data preparation**: Download and cleaning the data
2. **Data exploration & analysis**: Basic data exploration and advanced data visualization
3. **Model development**: Multiple distinct approaches for model engineering
4. **Model selection**: Based on model performance KPIs: \(AUC\), and \(Kappa\)
5. **Conclusion**: Final discussion/comments, including future work

Please note that the project comprises of a PDF file, containing the report, and the respective R and RMD files. All omitted code in this report can be found in detail in the R and RMD files.

1.1 Goals & Challenges

This capstone project has two main goals, described below:

- Evidence the data science concepts, knowledge, and skills acquired throughout the Professional Certificate Program (HarvardX) through their application to the **Machine Failures Prediction Data set.**
- Train and validate a models that predicts Machine Failures and their classification, using two methods.
- Demonstrate competence in several data science methods by training and validating a model that predicts Machine Failures, and their classification(s).

Two KPIs will be used to assess model performance, the \(Kappa\) and the \(AUC\). The goal of this project is to set a base line the for each of the two modelling algorithms for each of the two target column in the data set, and then test 4 different techniques for treating outliers in a data set, up-sampling, and the use of class weights on one of the two modelling algorithms \(AUC\) will be used as the primary matrix used to measure the performance of two class modelling engine and \(Kappa\) for the mutli-class classification engine. Once baselines have been reset, treatment of outliers will be applied, up-sampling and using class-weight will be applied to the data set and modelling algorithms, in an attempt to improve upon the original baselines, the models will be down selected based on being achieving the best score. Once the models have been down selected and tuned the models that achieved the highest scores for the respective primary metric will trained and tuned using and training & validation data set. Conclusions drawn regarding the success of these models will be based on the models used make prediction against the validation data set.

1.2 The Machine Failure Data set

The Machine Failure data set was obtained in the Kaggle (https://www.kaggle.com/) data base. The Machine Failure data set (https://www.kaggle.com/shivamb/machine-predictive-maintenance-classification) consists of a synthetic data set that reflects real predictive maintenance encountered in the industry to the best the publishers knowledge.

The data represents several key process variables that have been generated during the manufacturing process. The data set also includes information regarding which products were manufactured, whether any process-related failures occurred, and how these failures were classified.

2. DATA ANALYSIS

The following section details the data preparation and analysis process. Initially, a review of the available data sets was conducted, and an appropriate data set was selected. Once acquired, the data set was examined to inform an understanding of the information contained therein, and a hypothesis regarding which model would be the best fit for modelling was generated.

Additionally, the data visualization of the Machine Failure data set is going to create insists into the data set which will act a foundation or the decision-making process.

2.1 Acquire Data

As explained previously, data for this project was downloaded in the Kaggle website - link to the Machine Failure dataset (https://www.kaggle.com/shivamb/machine-predictive-maintenance-classification). The data set was replicated on GitHub (https://github.com/xxxxxxx/EdX_Capstone_Machine%20Failure), to allow all the project files to be saved in the same hosting server.

The first section of the R script downloads the Machine Failure data set in .csv format from GitHub and loads it into a data frame of raw data. Data will only be downloaded if not present in the current R working directory, to save the script´s running time.

2.2 Data Recode

The raw data stored in the **Machine Failures** data frame required transformation to fit the next phase of data exploration and visualization. The list of data transformation performed in the **Machine Failures** data frame is as follows:

- **Change column names**: Make the **Machine Failures** data set more readable by removing spaces in column names.

Recoded Columns Names

Original Column Names	New Column Names
Air.temperature..K.	Air_temperature_K
Failure.Type	Failure_Type
ï..UDI	Process_temperature_K
Process.temperature..K.	Product_ID
Product.ID	Rotational_speed_rpm
Rotational.speed..rpm.	target
Target	Tool_wear_min
Tool.wear..min.	Torque_Nm
Torque..Nm.	Type
Type	UDI

- **Change Data Types**: This is done to make sure the modelling algorithms know to to correctly handle and models the data that will be supplied to them.

Recoded Column Data Types

	Original Data Type	New Data Type
UDI	integer	integer
Product_ID	character	factor
Type	character	factor
Air_temperature_K	numeric	numeric
Process_temperature_K	numeric	numeric
Rotational_speed_rpm	integer	integer
Torque_Nm	numeric	numeric
Tool_wear_min	integer	integer
target	integer	factor
Failure_Type	character	factor

- **Categorical variables**: Remove spaces in strings/characters columns. Below are the detail steps in R for the initial data transformation phase.

Recoded Factor Levels For Failure Type

Original Factor Levels	New Factor Levels
Heat Dissipation Failure	Heat.Dissipation.Failure
No Failure	No.Failure
Overstrain Failure	Overstrain.Failure
Power Failure	Power.Failure
Random Failures	Random.Failures
Tool Wear Failure	Tool.Wear.Failure

2.3 Data Cleanse

- **Discrepancy in Data**: Conduct a basic data check to ensure that the data follows a set of logical assumptions. It is worth noting that a minor discrepancy in the date was identified, which could affect the model's performance. Below is the summary count of target grouped by failure type. On inspection, it appears that the two class response ("target") column for the process failure has contradictory labeling for No.Failures with 9 rows in the target column stating that a failure had occurred but the Failure Type states No Failure this is in contradiction to the rest of the data set and will be removed.

Summary Count of Target group by Failure Type

target	Failure_Type	n
0	No.Failure	9643
0	Random.Failures	18
1	Heat.Dissipation.Failure	112
1	No.Failure	9
1	Overstrain.Failure	78
1	Power.Failure	95
1	Tool.Wear.Failure	45

Summary Count groups that logically seemed to be mislabelled

target	Failure_Type	n
0	No.Failure	9643
0	Random.Failures	18
1	No.Failure	9

Due to concerns about confusion and complications during the data modelling process, this logical discrepancy in the data was resolved to ensure the best possible outcome. After considering numerous treatment methods for dealing with discrepancy, removing the contradicting rows seemed the most appropriate option.

For simplicity the two class response column factors have been recoded from 0 & 1 to Pass and fail.

Summary Counts of target and Failure types are data correction

target	Failure_Type	n

Target	Failure Type	9643
0	Random.Failures	18
1	Heat.Dissipation.Failure	112
1	Overstrain.Failure	78
1	Power.Failure	95
1	Tool.Wear.Failure	45

2.4 Data Summary

The original data set consists of 10 000 data points stored as rows with 14 features in columns.

The data set consists of 10 000 data points stored as rows with fourteen features in columns • UID: unique identifier ranging from 1 to 10000

• productID: consisting of a letter L, M, or H for low (50% of all products), medium (30%), and high (20%) as product quality variants and a variant-specific serial number

• air temperature [K]: generated using a random walk process later normalized to a standard deviation of 2 K around 300 K

• process temperature [K]: generated using a random walk process normalized to a standard deviation of 1 K, added to the air temperature plus 10 K.

• rotational speed [rpm]: calculated from power of 2860 W, overlaid with a normally distributed noise

• torque [Nm]: torque values are normally distributed around 40 Nm with an $\ddot{I}f$ = 10 Nm and no negative values.

• tool wear [min]: The quality variants H/M/L add 5/3/2 minutes of tool wear to the used tool in the process.

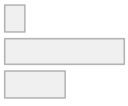
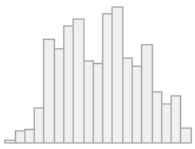
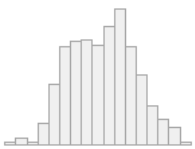
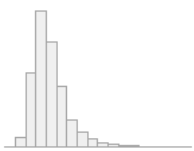
• target: two class response column to indicate if a failure occurred during the process.

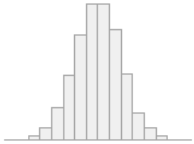
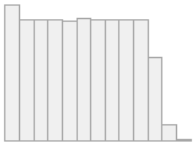
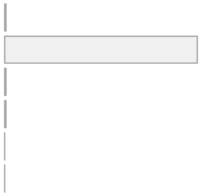
• Failure Type: label that indicates, whether the machine has failed in this particular data point for any of the following failure modes are true.

Data Frame Summary

MM

Dimensions: 9991 x 10
Duplicates: 0

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
UDI [integer]	Mean (sd) : 5000.2 (2887.3) min < med < max: 1 < 5000 < 10000 IQR (CV) : 5001 (0.6)	9991 distinct values		0 (0.0%)
Product_ID [factor]	1. H29424 2. H29425 3. H29432 4. H29434 5. H29441 6. H29452 7. H29457 8. H29462 9. H29466 10. H29481 [9990 others]	1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 1 (0.0%) 9981 (99.9%)		0 (0.0%)
Type [factor]	1. H 2. L 3. M	1002 (10.0%) 5996 (60.0%) 2993 (30.0%)		0 (0.0%)
Air_temperature_K [numeric]	Mean (sd) : 300 (2) min < med < max: 295.3 < 300.1 < 304.5 IQR (CV) : 3.2 (0)	93 distinct values		0 (0.0%)
Process_temperature_K [numeric]	Mean (sd) : 310 (1.5) min < med < max: 305.7 < 310.1 < 313.8 IQR (CV) : 2.3 (0)	82 distinct values		0 (0.0%)
Rotational_speed_rpm [integer]	Mean (sd) : 1538.8 (179.3) min < med < max: 1168 < 1503 < 2886 IQR (CV) : 189 (0.1)	941 distinct values		0 (0.0%)

Variable	Stats / Values	Freqs (% of Valid)	Graph	Missing
Torque_Nm [numeric]	Mean (sd) : 40 (10) min < med < max: 3.8 < 40.1 < 76.6 IQR (CV) : 13.5 (0.2)	577 distinct values		0 (0.0%)
Tool_wear_min [integer]	Mean (sd) : 107.9 (63.7) min < med < max: 0 < 108 < 253 IQR (CV) : 109 (0.6)	246 distinct values		0 (0.0%)
target [character]	1. Fail 2. Pass	330 (3.3%) 9661 (96.7%)		0 (0.0%)
Failure_Type [factor]	1. Heat.Dissipation.Failure 2. No.Failure 3. Overstrain.Failure 4. Power.Failure 5. Random.Failures 6. Tool.Wear.Failure	112 (1.1%) 9643 (96.5%) 78 (0.8%) 95 (1.0%) 18 (0.2%) 45 (0.5%)		0 (0.0%)

The summary shows that for the 10 columns and 9,991 observations that there is no duplication of data and no missing values. The UDI and Product ID are unique for every observation. The distribution for the key process variables seems to be fairly well normally distributed, with rotational speed being slightly shewed to the left had side.

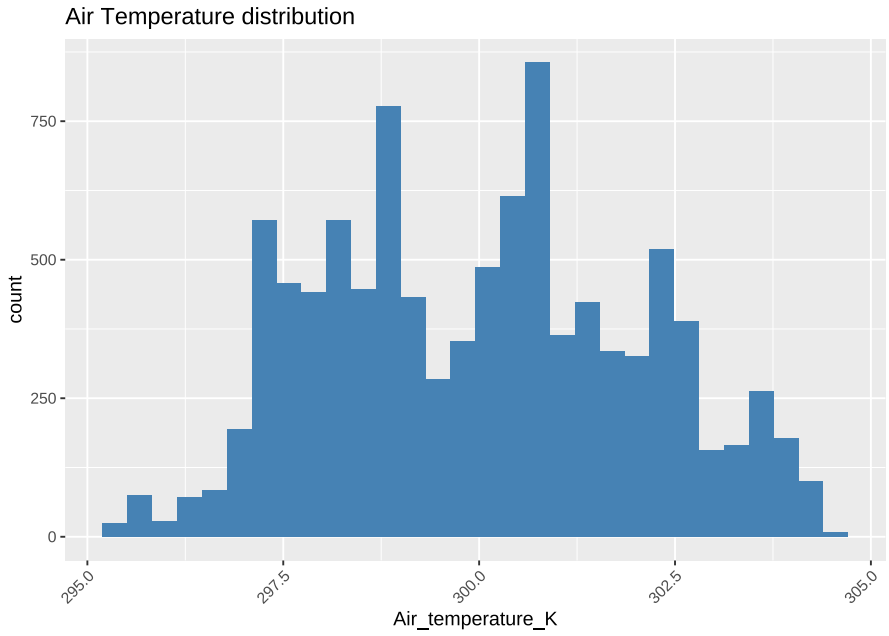
The target columns and failure type columns shows that process failures occur 3.5% of the time making occurrences of failure rather rare for the overall process. When we look at the specific failure types it can be seen that most of failure occurrences of less than one percent. This will most likely mean that performance matrices of RMSE and Accuracy will not be the best measures by which to access model performance, selected metrics will be discussed in a later section.

2.5 Data Exploration

This section focuses on visualizing and understanding what how different process variables impact the different failure types that we need to classify or predict. Initially this section looks at each individual process variables and how these process variables influence or are influenced by product type and failure type.

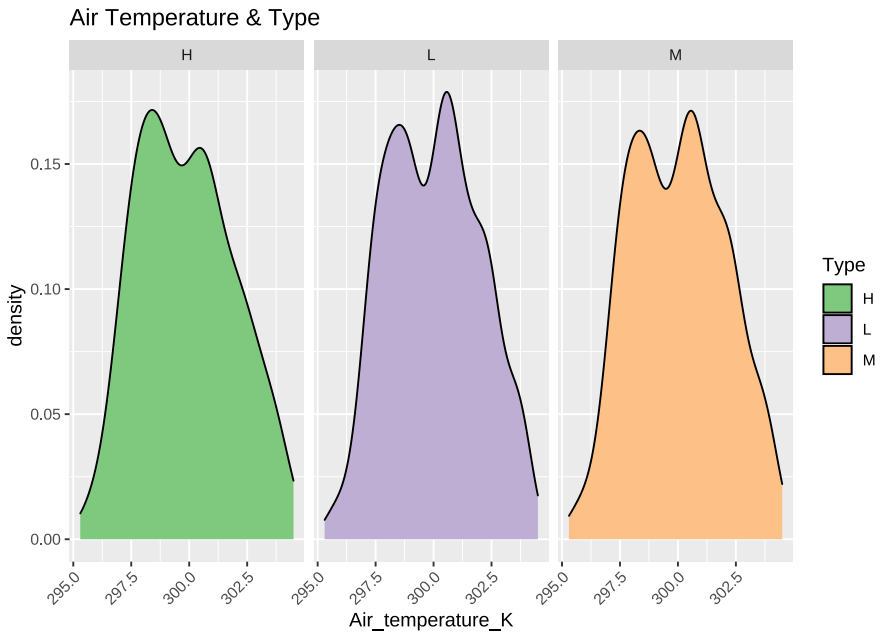
Air Temperature:

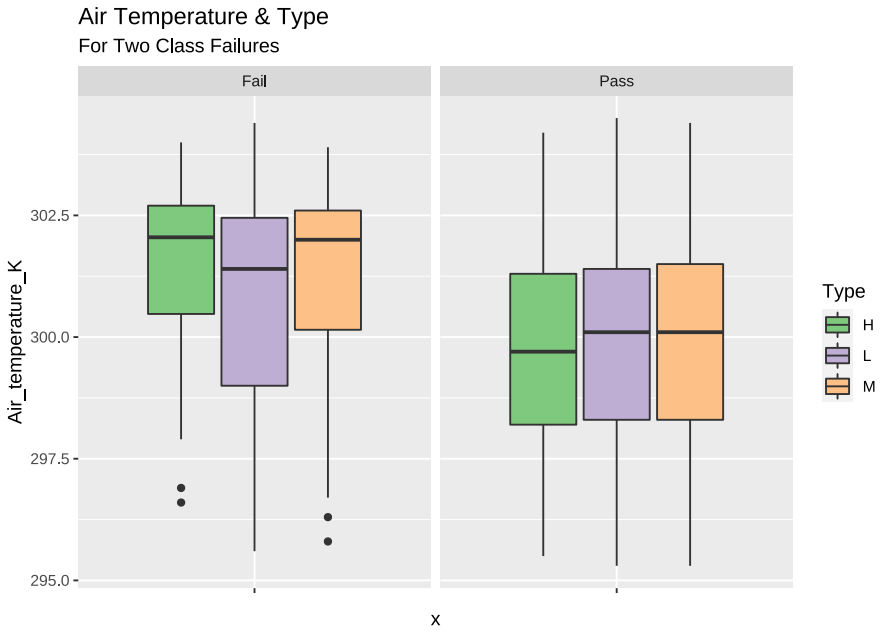
Air temperature is more of an environmental variable than a process variable and can be much harder to control.

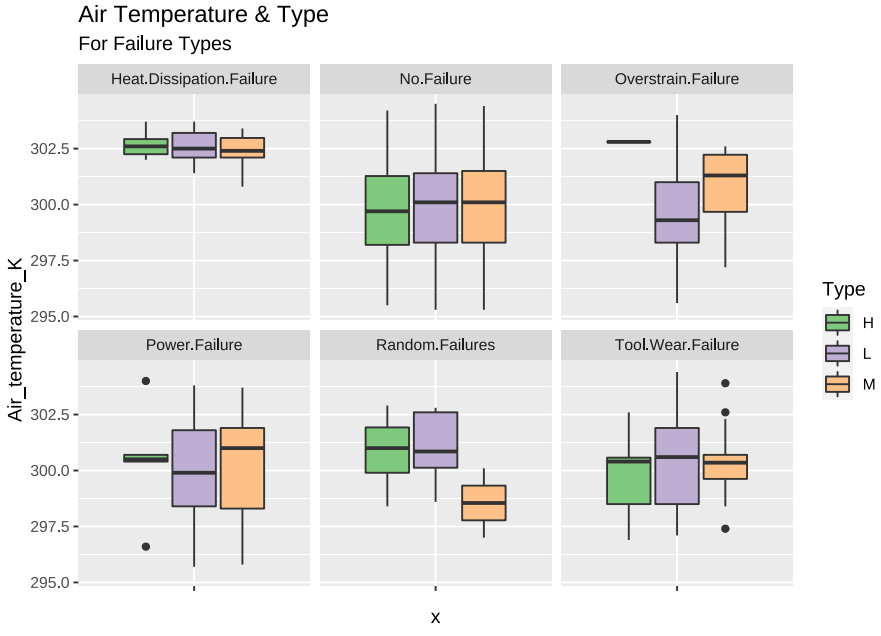


Check of Air Temperture Skewness

	x
Air_temperature_K	0.114





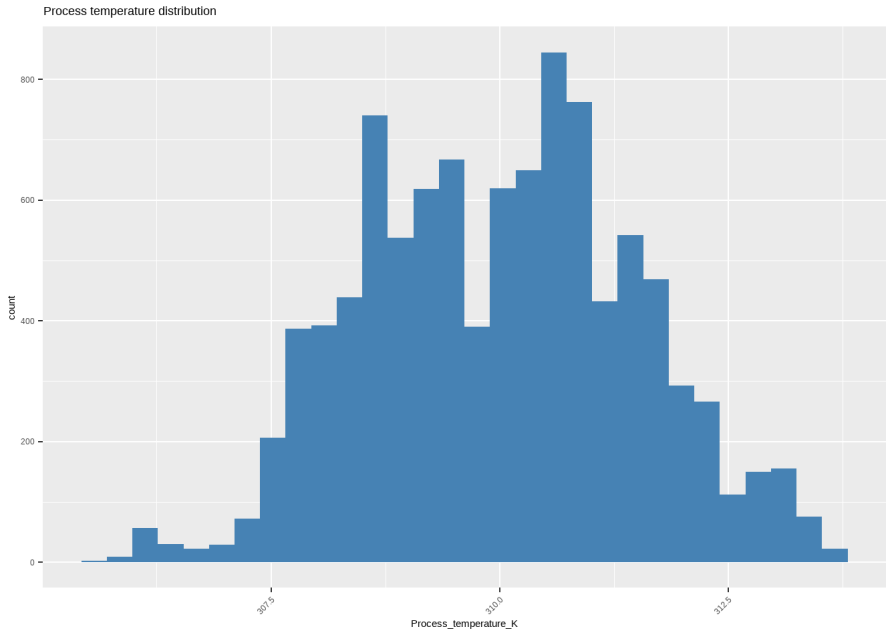


Key takeaways Air Temperature:

- The histograms show that the air temperature follows an approximate normal distribution centered around 300 K
- The density plots for the 3 product types are very similar in their shape, with H type profile is slightly less pronounced then the L & M. The density plots clearly show that product types are exposed the same air temperatures during production.

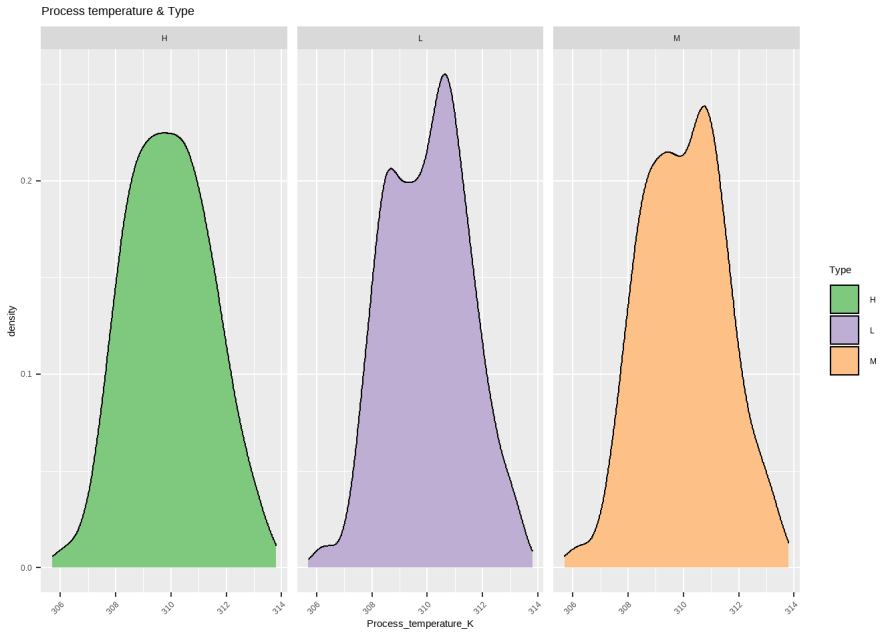
- The Failure type box plots allow us to draw some simple conclusions, Heat Dissipation failures are more likely to occur at higher Air Temperature for all product types. Shows that random failure for type M are more likely to occur at lower temperature than for H & L. The box plots all show a very narrow range for which Power Failures & Over strain failure occur for Type H products.

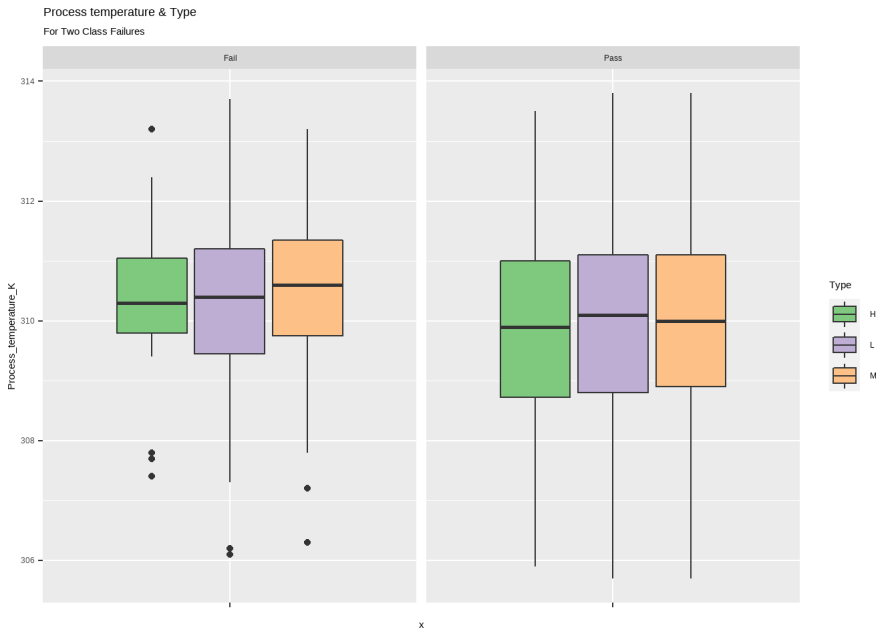
Process Temperature

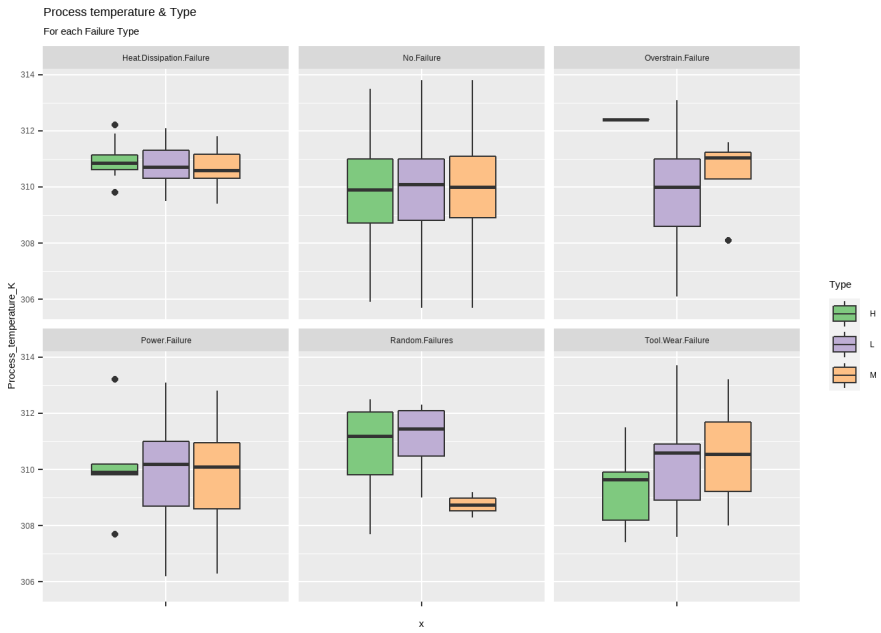


Check of Process Temperture Skewness

	x
Process_temperature_K	0.015





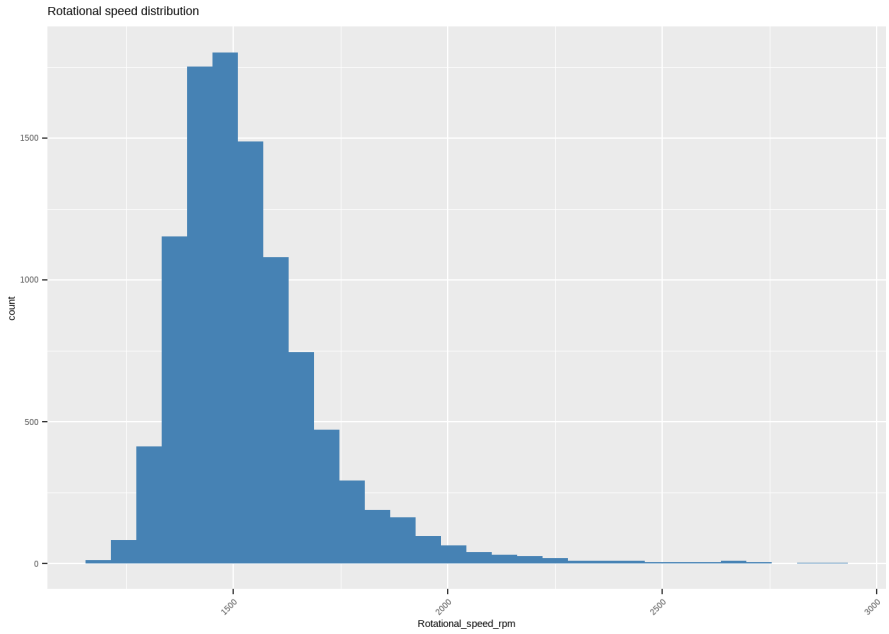


Key takeaways Process Temperatures:

- The histograms show that the process temperature follows an approximate normal distribution centered around 310 K

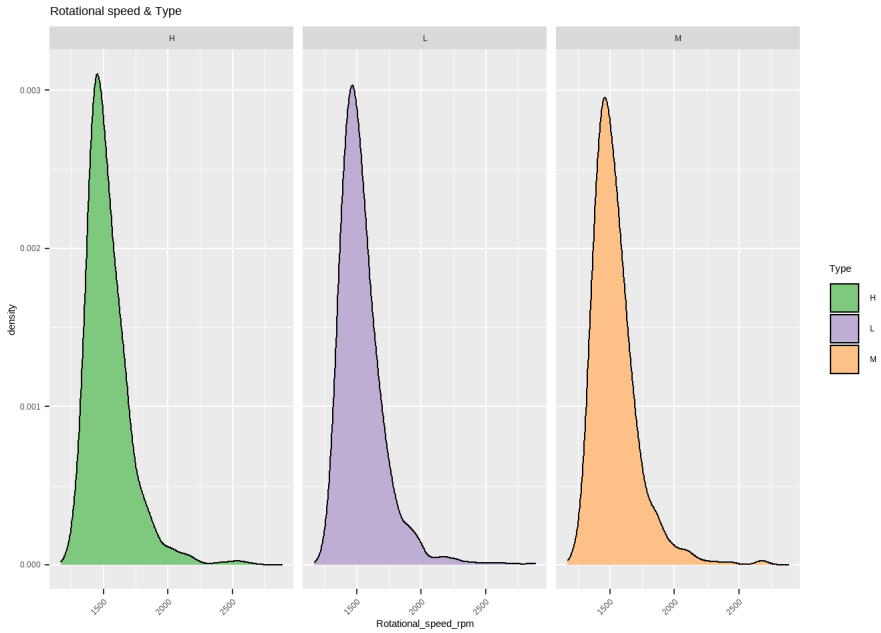
- The density plots for the 3 product types show that type L product has a very visible density peck at around 311'k this peck can be seen in type M but the peak is less distinct and there is no visible peak at all for M. It seems that the process temperature for L & M type products is hotter than M
- The box plots are show some really narrow ranges of process temperatures for when heat dissipation failure occur for all product types. Again with random failure we see a very narrow range with short whiskers for Type M which is different to the range for L & H which themselves are very similar.

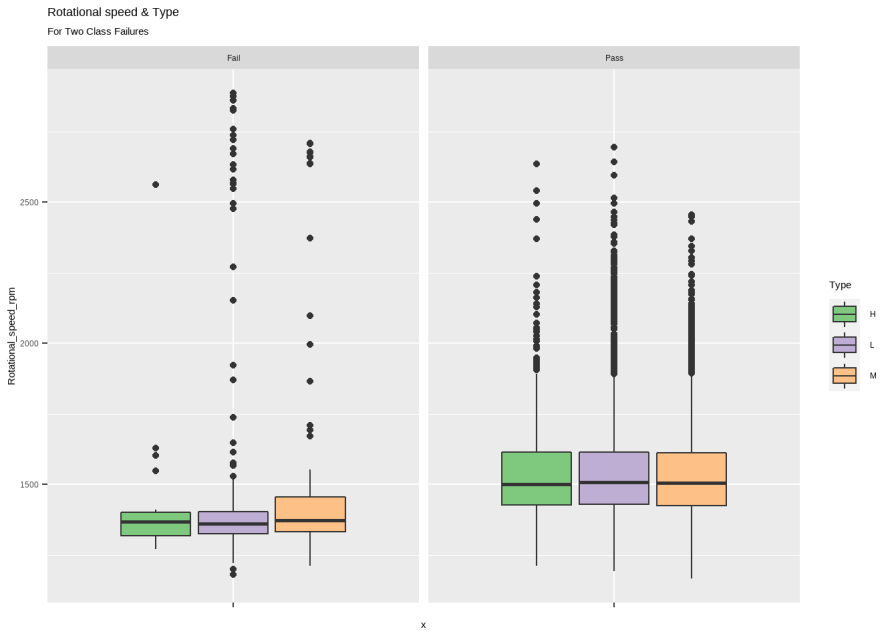
Rotational Speed

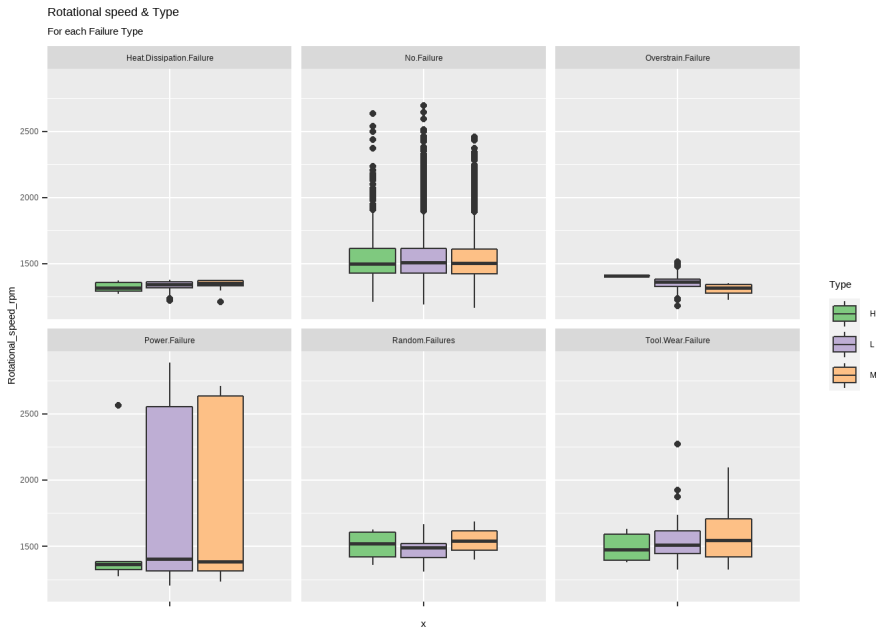


Check of Rotational Speed Skewness

	x
Process_temperature_K	0.015





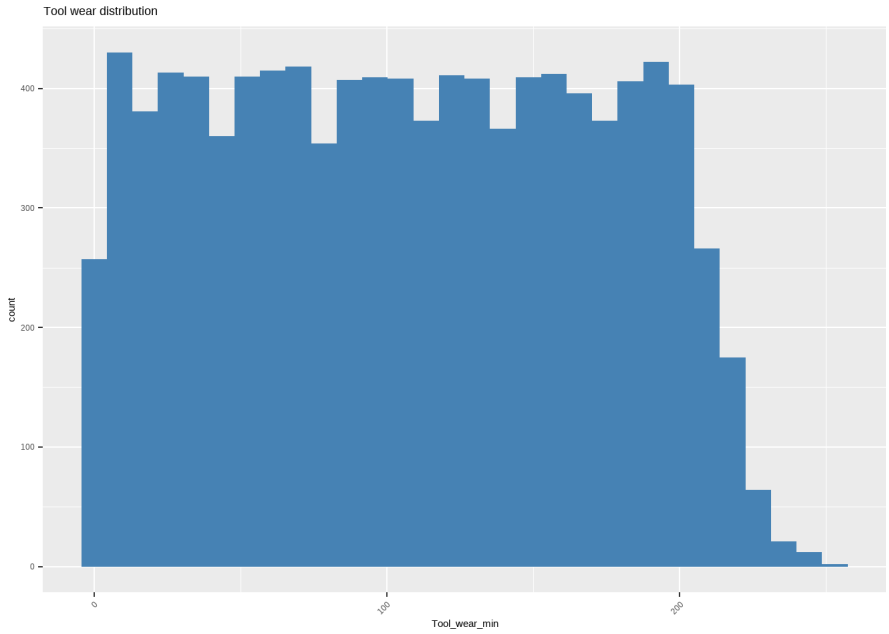


Key takeaways Process Temperatures:

- The histograms show that the rotational has a slight shrew towards the left hand side of the plot.
- The density plots clearly show that there is little difference in the rotational speeds used across the 3 product types.

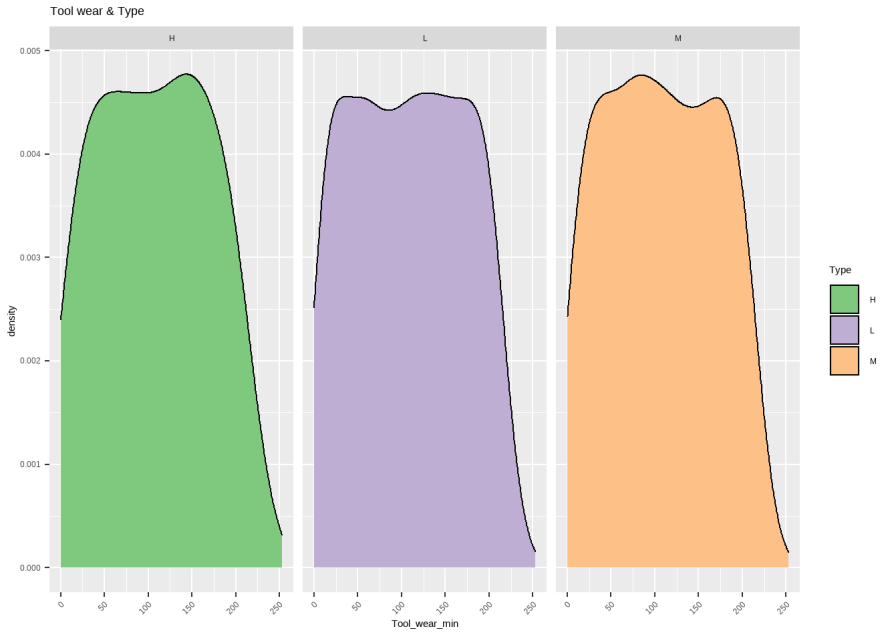
- The box plots for Heat Dissipation, Over-strain, Random Failure are very narrow with smaller whiskers. Heat Dissipation & over-strain interquartile ranges do not heavily overlap with the interquartile range of No Failure, while the interquartile ranges of random failures do. Power Failure has the largest range of the failure types in this plots with a very low median, showing that power failures happen at both high and low rotational speeds with a tendency to occur more at low speed. Tool wear also has a narrow range with the median being towards the low ranges of rpm.
- The box plots also indicate that there are a number of statistical outliers predominately for No.Failure classification.

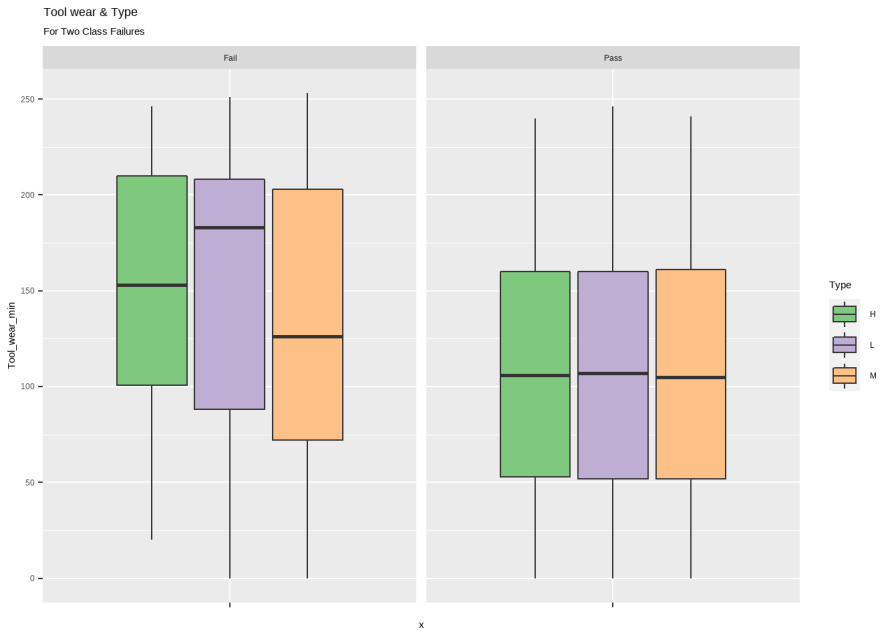
Tool Wear

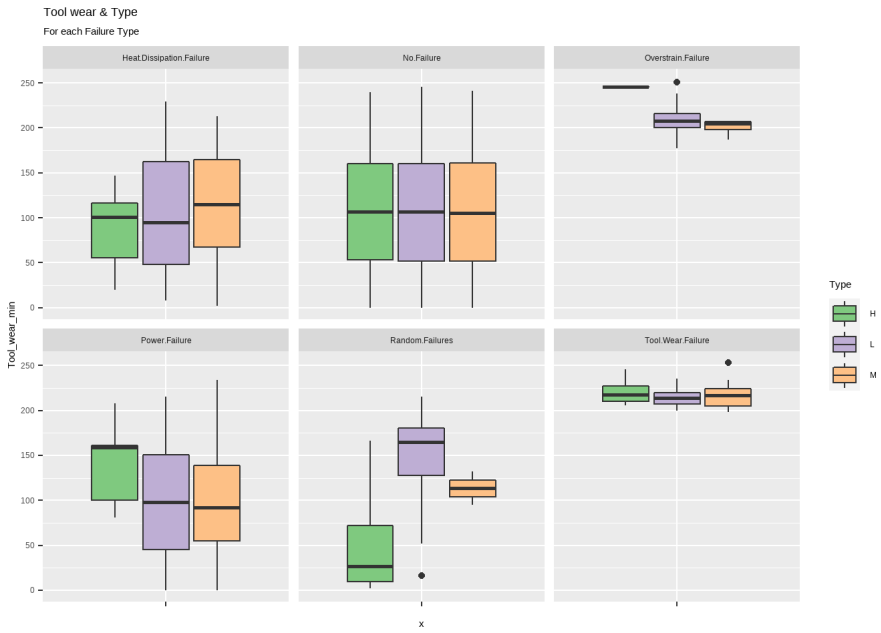


Check of Tool wear Skewness

	x
Process_temperature_K	0.015







Key takeaways tool wear:

- The histograms shows that tool life very quickly drops off after 200 mins of use, by the steep line at the right side.
- The density plot clearly show that there is little difference in the tool wear across the 3 product types, it seems that the right hand slope of the density plots is steeper for product H than it is for L & M

- The ranges of Heat dissipation, Power failure overlap those of No failure, Overstrain & Tool wear failures have narrow ranges with short whiskers in either direction with no overlap of the interquartile range of No Failure.

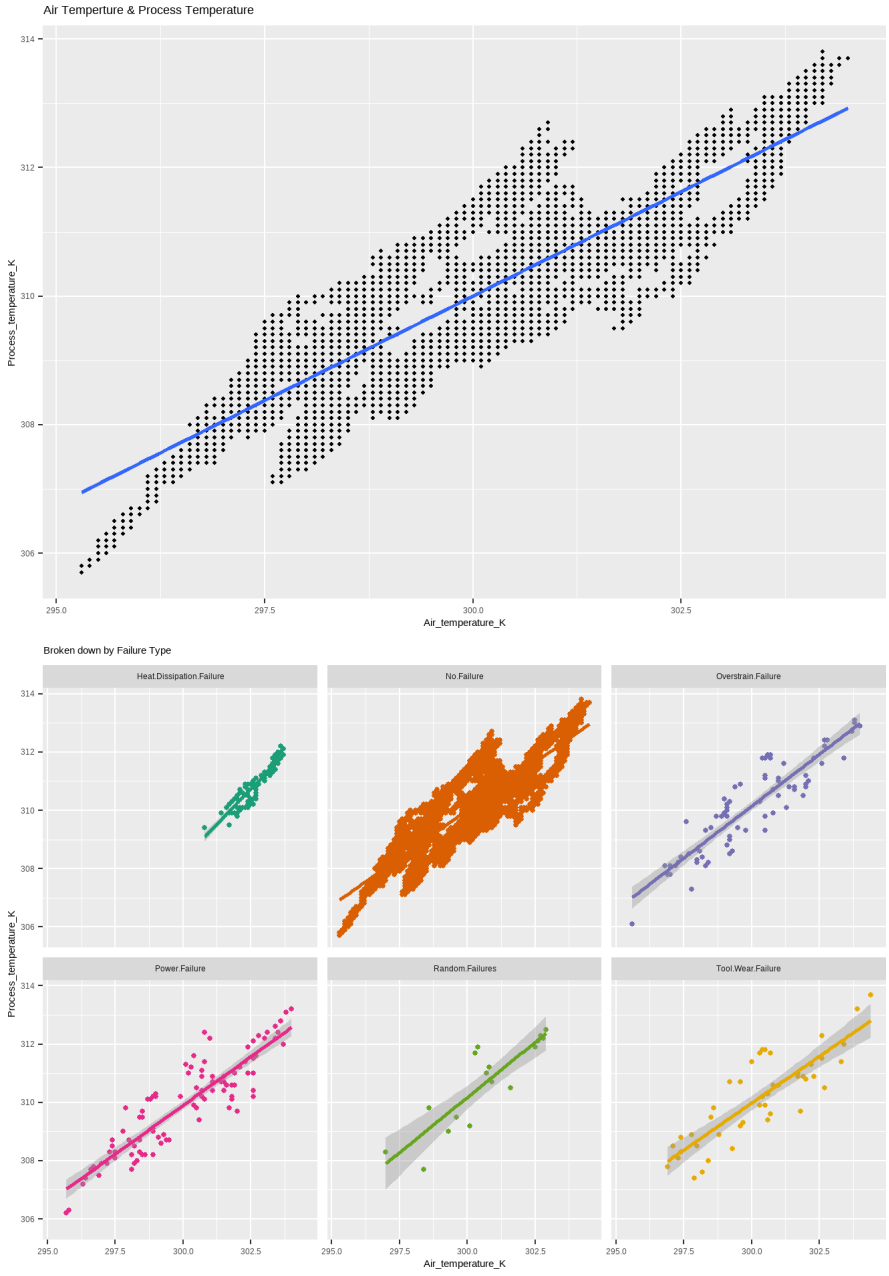
After reviewing the process variable is isolation, from the density plots it can be seen that the process variables Air Temperature, process Temperature, tool wear, rotational speed and torque used to manufacture the different product type are similar.

From the box plots it is possible to see what ranges different process failures occur for different product types. In a number of previous box plot it was possible to see what values for the process variable would likely result in some form of failure.

More Exploration:

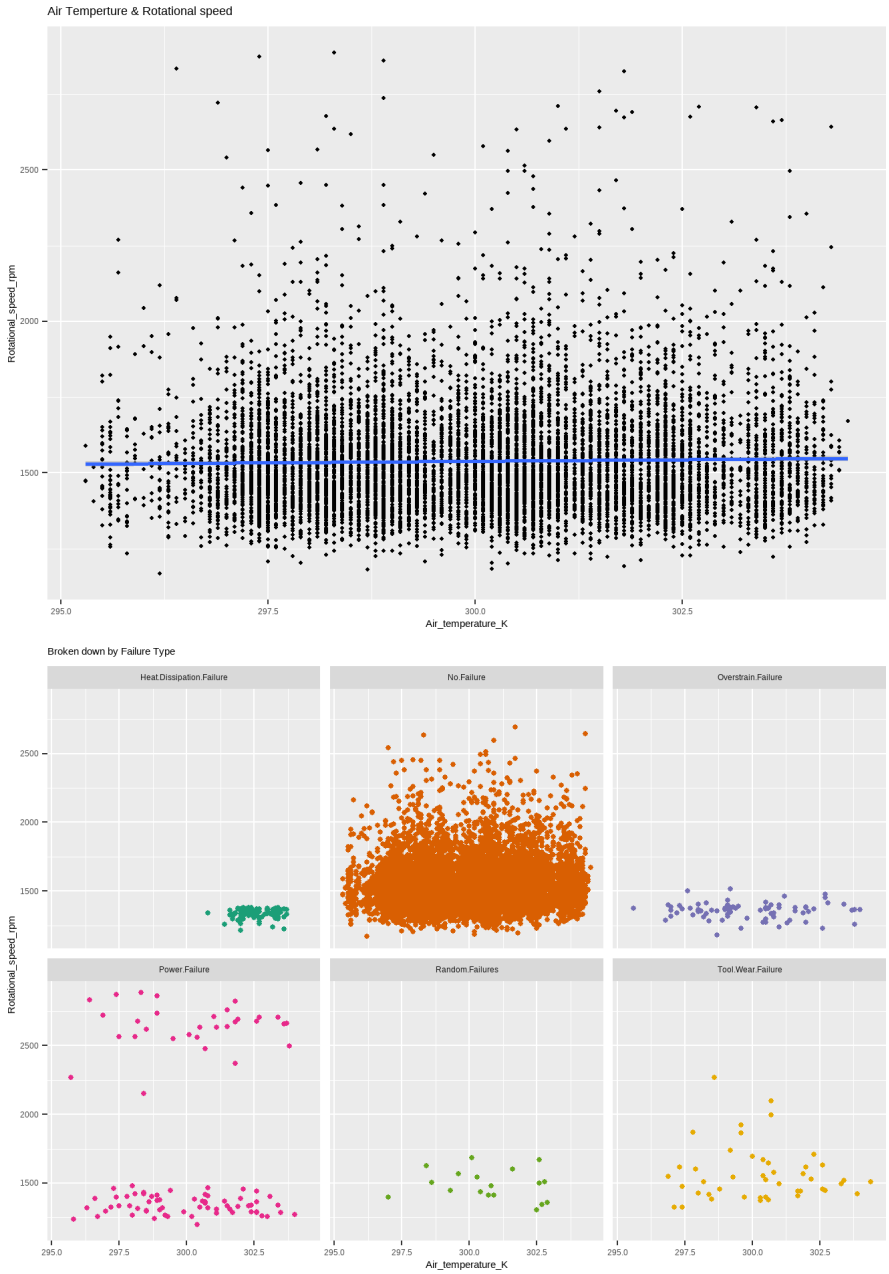
The aim of the section below is to further the understanding gained from the previous section and this time instead of looking at the process variable in isolation we shall a look how combinations of two different process variables relate to the different failure modes that occur during the manufacturing process.

Air Temperature & Process Temperature



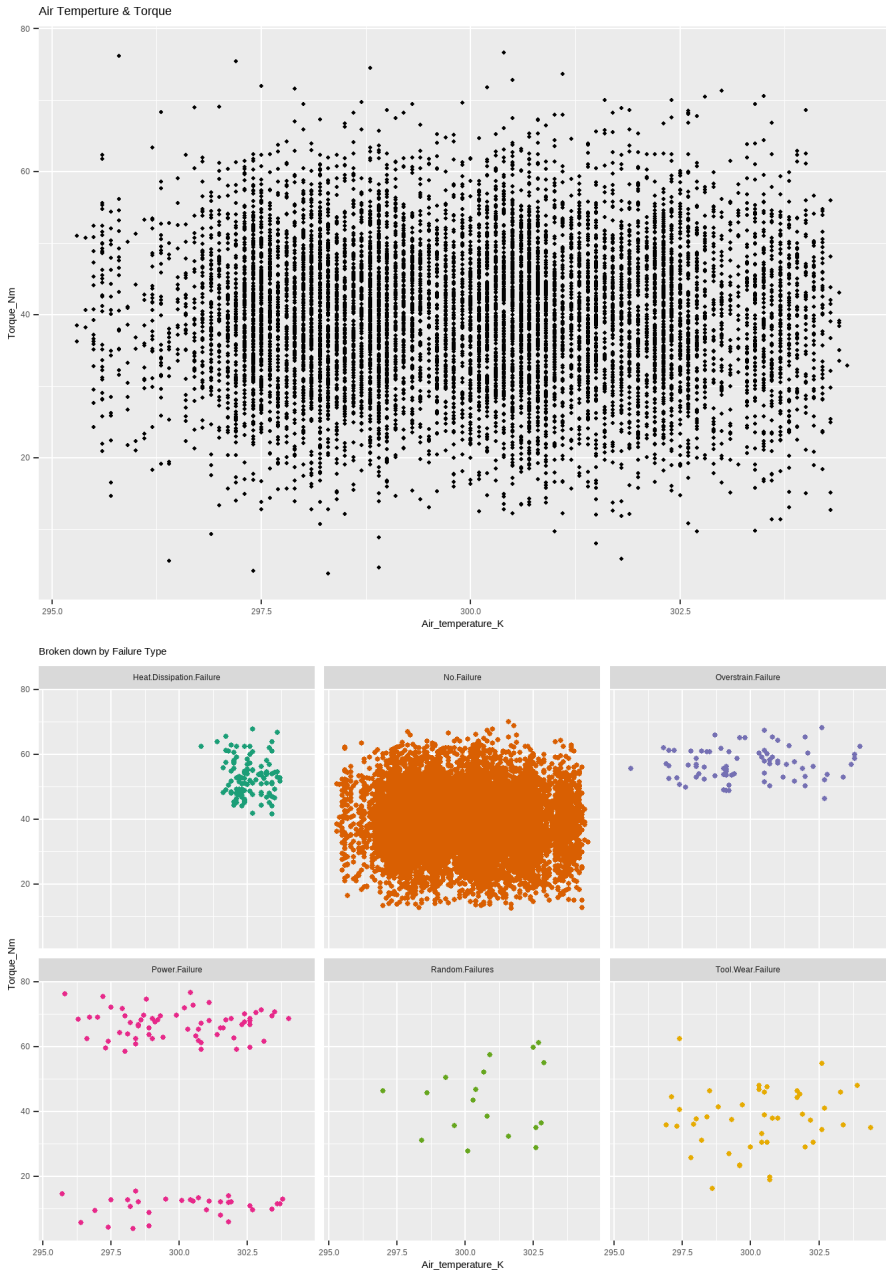
As can be seen from the charts air temperature has a strong correlation to process temperature, this correlation is strongest for Heat Dissipation, and weaker for random & tool wear failures. Heat dissipation failure are all tightly cluster in the top right-hand side of the failure charts. Meaning high air temperature and high process temperatures are ideal conduction for a heat dissipation failure to occur.

Air Temperature & Rotational speed



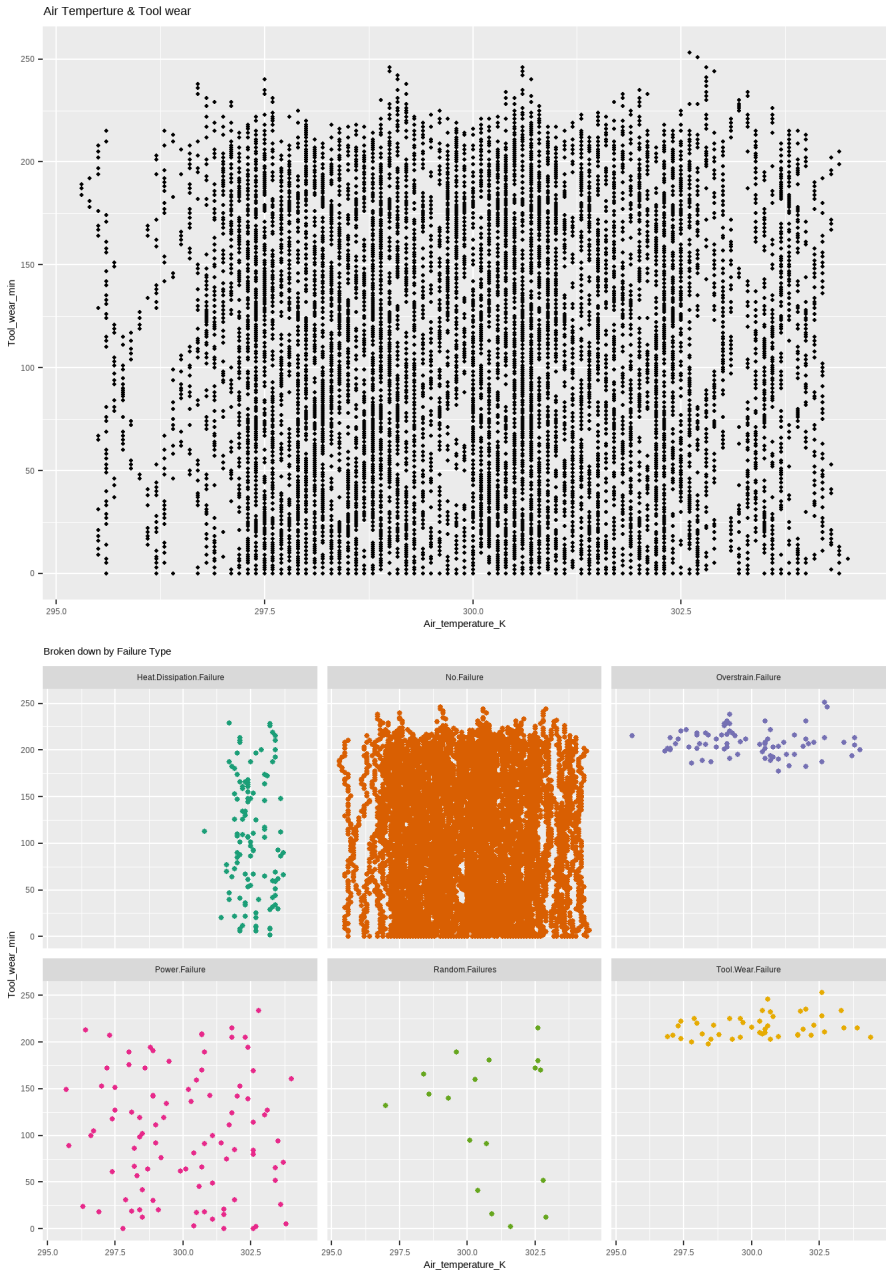
Air Temperature & Rotational speed do not have a linear correlation, but the failure type scatter plots have again provided some really nice distinctive clusters of when the failure occur in relation to Air Temperature and Rotational speed. the data point clustering for tool wear is quite large compared to the other failure modes, but it still easy to see what failure can occur and will not occur.

Air Temperature & Torque



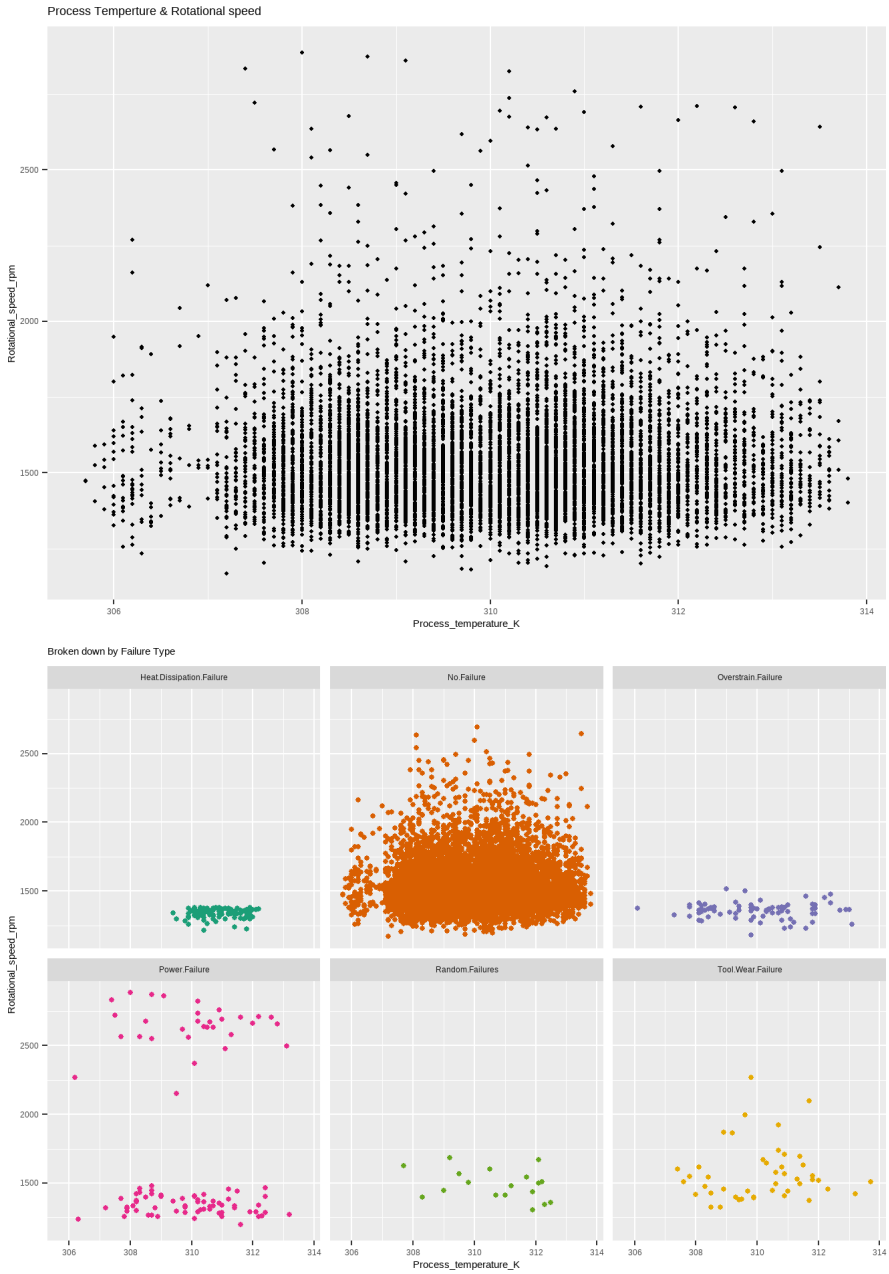
Again, it is possible to see obvious data point cluster for each of the failures mode in relation to Torque and Air Temperature. As with the previous plot in this section Heat dissipation power failure and over-strain data points have all created quite easy to see data grouping. Random and Tool wear failures do not seem to want to create nice tight groups like the others, the grouping themselves do not seem to significantly overlap.

Air Temperature & Tool wear



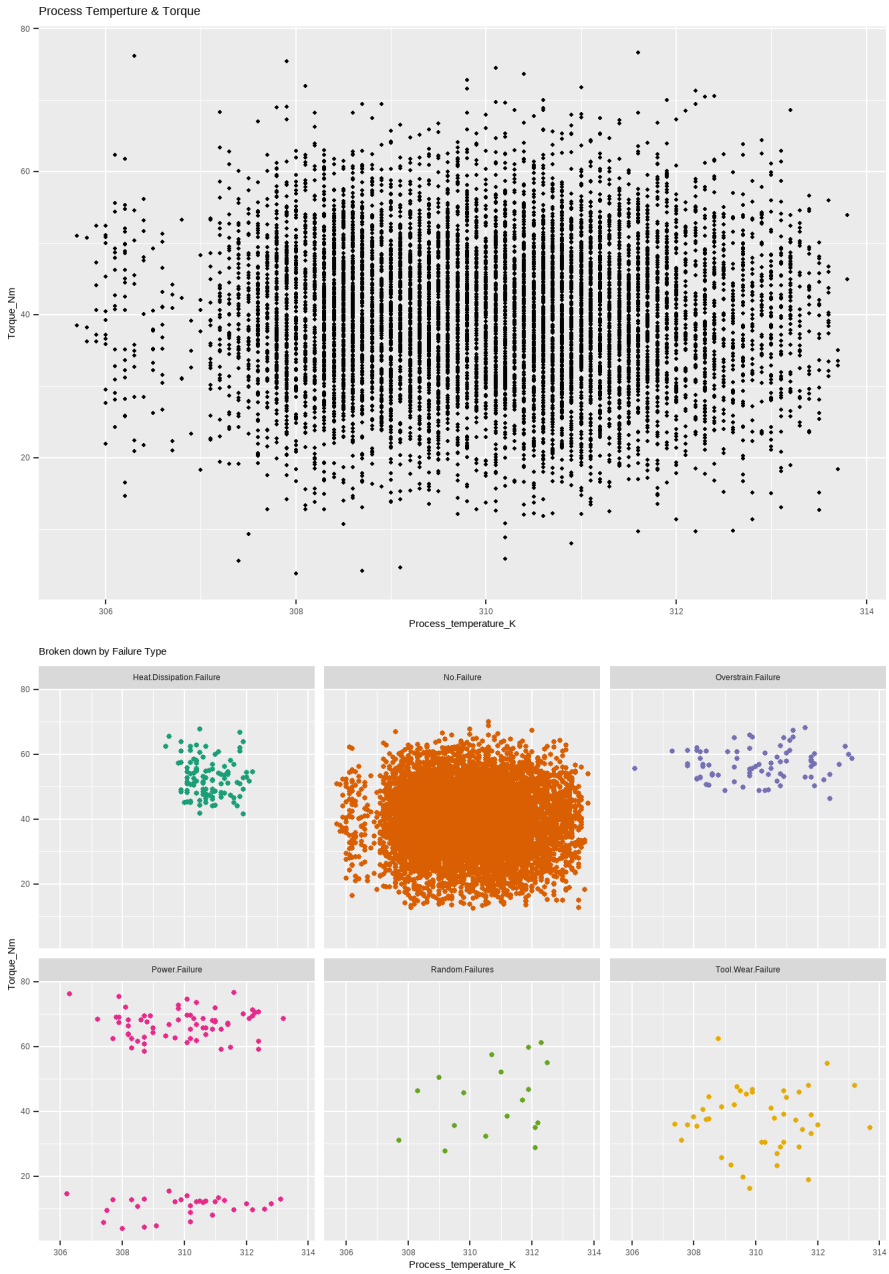
In these plots the data points for tool wear failure have all created a grouping at the top of the charts, which is the first time we have seen a nice tight clustering of its data points. Other previous grouping is still visible although not as densly grouped together as previous plots. Power failure clearly showing that Tool & Air Temperature do not correlate for the occurrence of the failure mode.

Process Temperature & Rotational_speed_rpm



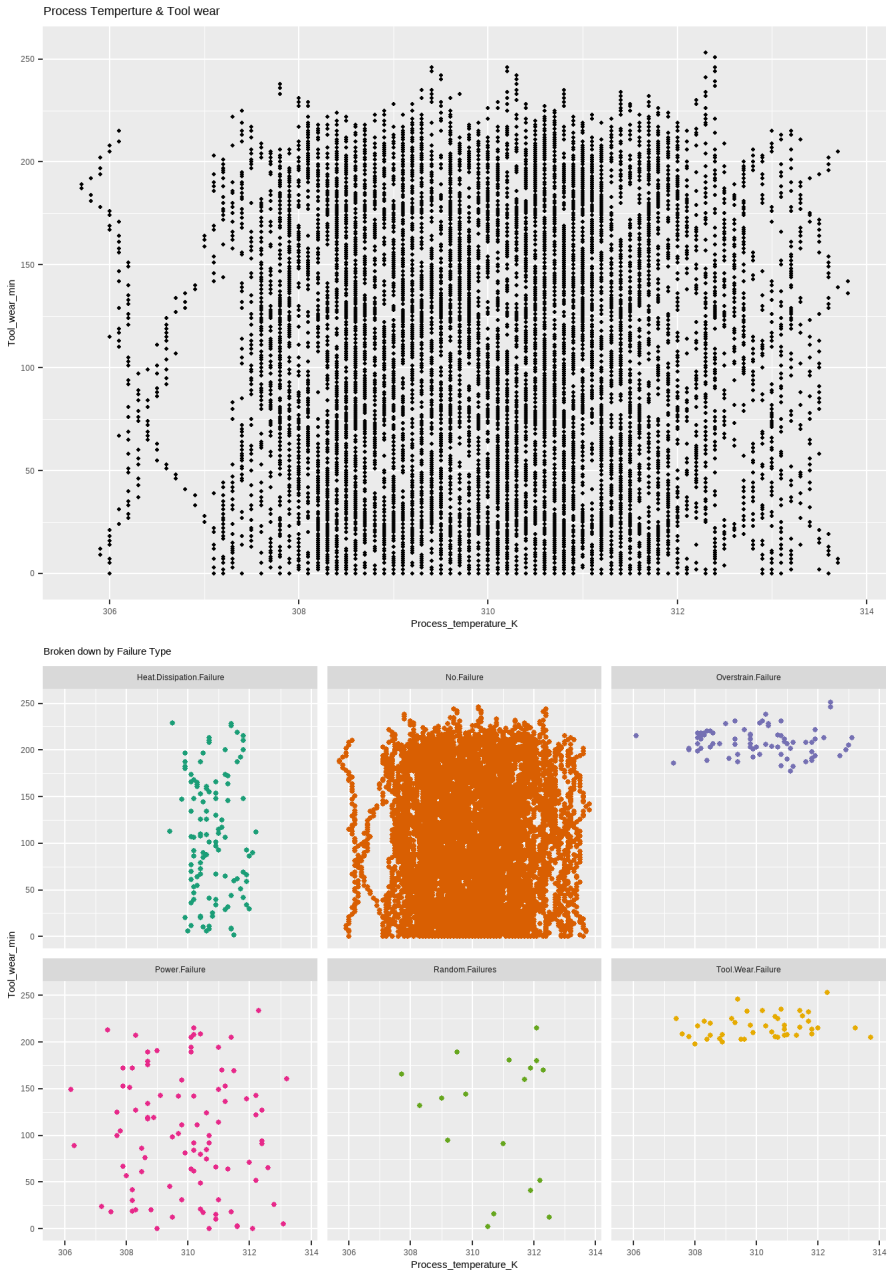
As with previous plot we are seeing very obvious data points clusters for various failure types.

Process Temperature & Torque



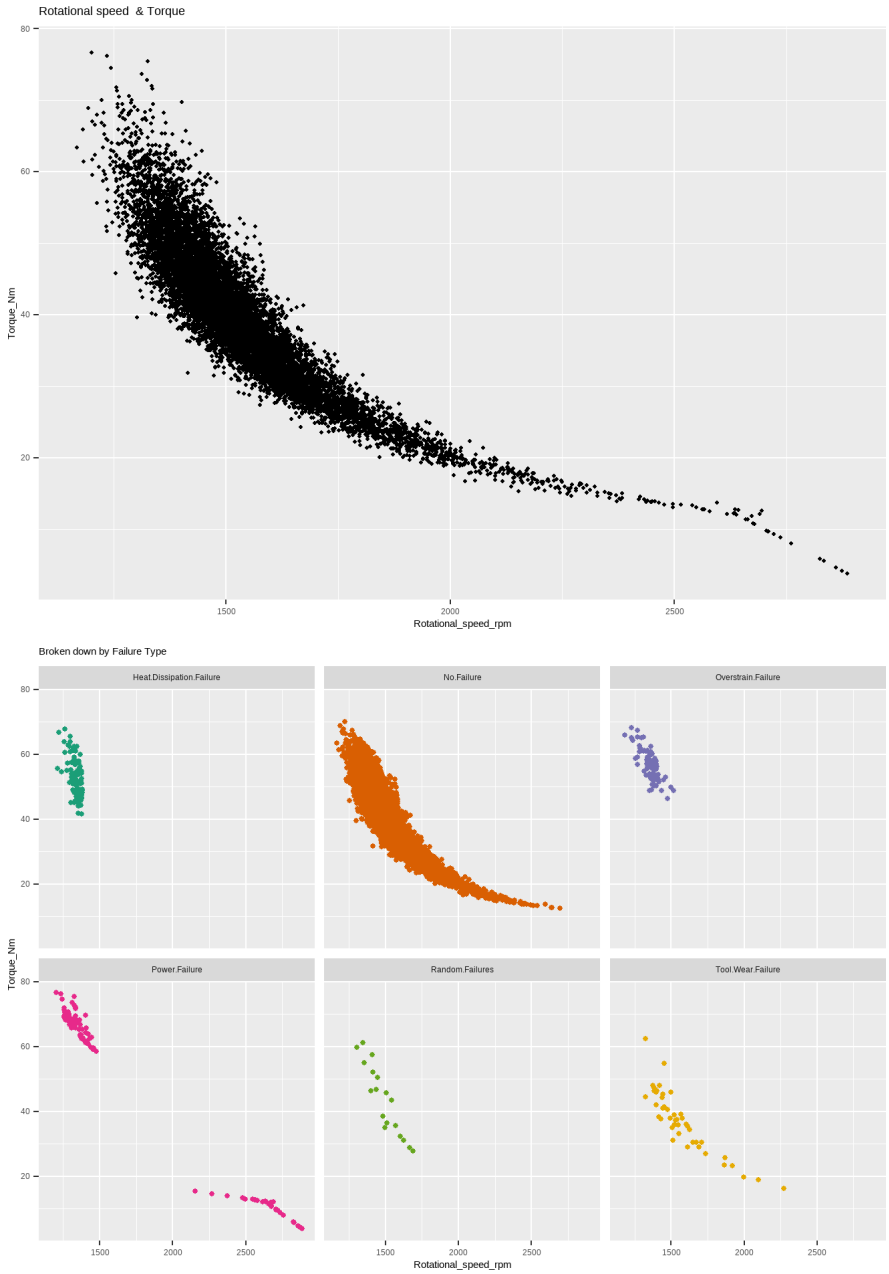
Again, very distinct groupings of data points for the failure types. With this plot like the others it is clear between what values different failure modes will and won't occur.

Process Temperature & Tool wear



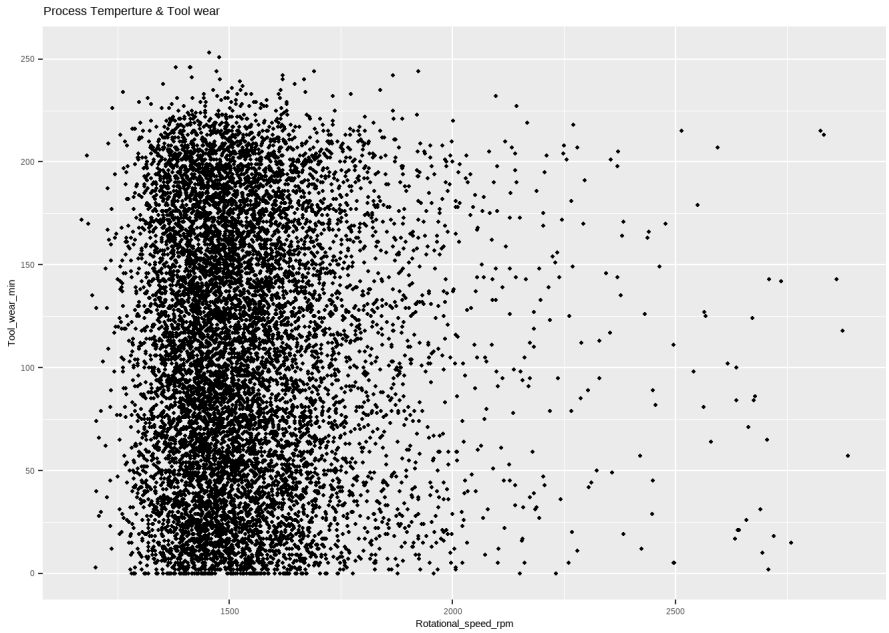
The data points have again created dense clearly visible groupings clearly showing what failure modes can happen between a range of process variables.

Rotational speed & Torque

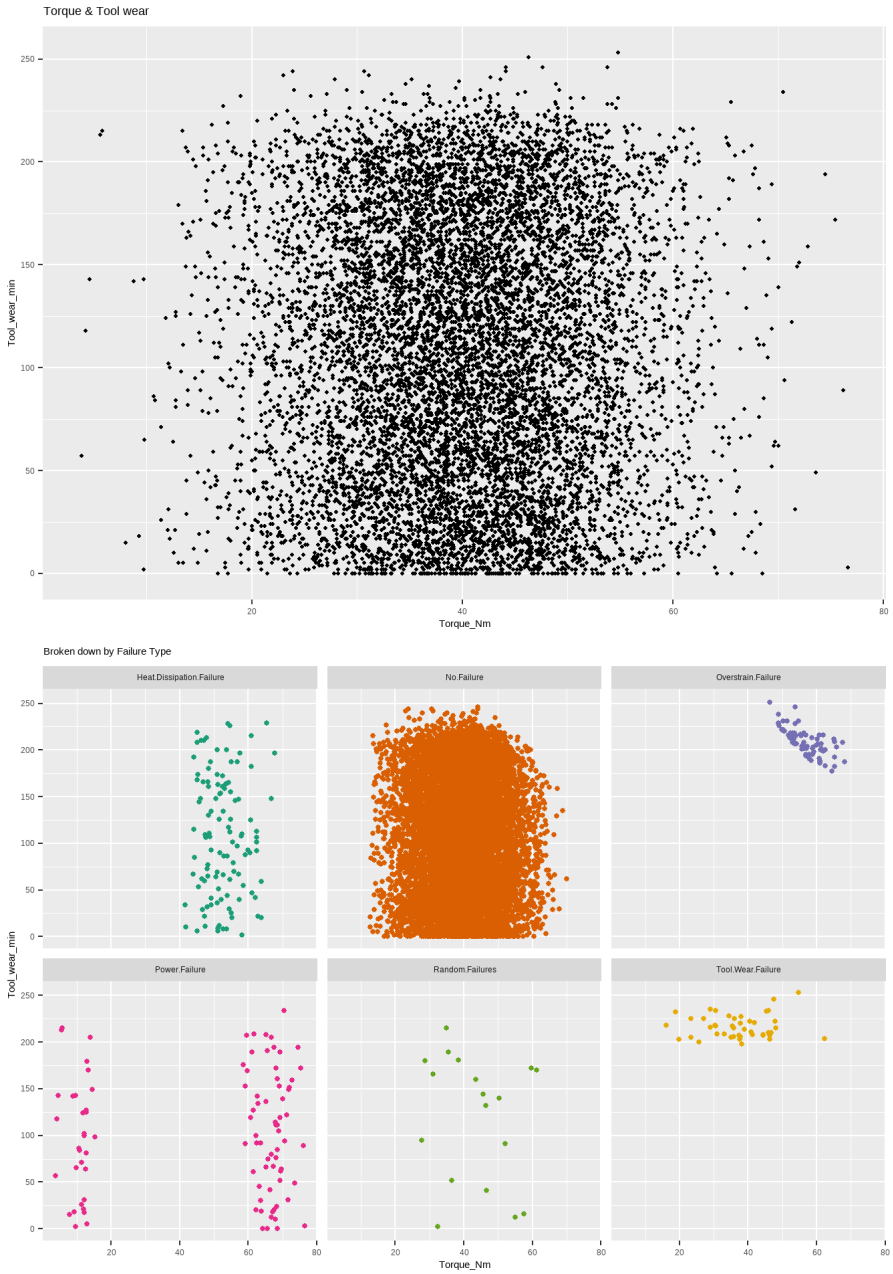


Contained data point grouping for the of the failure types, show the range for the two-process variable at which the failures occur.

Rotational speed & Tool wear



Torque & Tool Wear



By comparing all the different process variables against one another, it is possible to see the ranges in which these failure modes occur. The data clearly shows that the failures do concur with set parameter ranges with some overlap, modelling should be able to achieve some good results overall, misclassification of failure types is likely to occurs because of this overlap.

3 MODELLING

3.1 Partition Data

Next, the seed is going to be set to achieve repeatable results and the **Machine Failures** data set will be 60/20/20 split. This means 60% of the Machine Failures data frame is going to be used for the training data set, **train_data**, 20% to the testing data set, **test_data**, and 20% to the testing data set, **test_data.** The 60/20/20 split ratio has been chosen as the data set is deemed to be large enough. The creation of the testing and training data sets can be found to the script and rmd files.

3.2 Model Baselines

Two modelling algorithms were chosen for this project. They are caret – Ranger and caret – xgbTree. These two modelling algorithms are used for modelling both the two class, and mutli-class response columns.

At this stage each of the two techniques for each of the two target columns are trained. This will be used to create a baseline set of statistics that can be used as a reference points during the models development stage.

Two Class Model Baselines

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	F0.5	RSME	Overall_Error	AUC	F1	Balanced Accuracy	Accuracy
target	ranger	No-Sampling	No_Treatment	No_Weights	0.6956522	0.1466655	0.0215108	0.9751603	0.5981308	0.7400963	0.9784892
target	xgbTree	No-Sampling	No_Treatment	No_Weights	0.6302521	0.1565639	0.0245123	0.9768338	0.5504587	0.7239101	0.9754877

The baselines for the Two Class Classification are all pretty close, ranger seems to be performing slightly better at this stage when we look across the key measures the two measure where ranger is not leading is AUC, and LogLoss.

Multiclass Model Baselines

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	Overall_Error	AUC	Mean_Error	LogLoss	Accuracy	Kappa
Failure_Type	ranger	No-Sampling	No_Treatment	No_Weights	0.0250125	0.9084271	0.0083375	0.0831943	0.9749875	0.5296780
Failure_Type	xgbTree	No-Sampling	No_Treatment	No_Weights	0.0215108	0.9209022	0.0071703	0.0750712	0.9784892	0.6288141

The baselines for the Mutli Class Classification are also all pretty close, with ranger leading the way performing best for all measures.

3.3 Model Development

Here the aim is to improve upon the baselines we have already gathered. We shall look at removing outliers using a number of different group options as well as up sampling on the data to improve the prediction outcomes.

Here the aim is to improve upon the baselines we have already gathered. We shall look at removing outliers using four different group options as well as up sampling on the data to improve the prediction outcomes.

Outliears Treatments Explained:

The interquartilerange is the area between the 75^{th} and the 25^{th} percentile of a distribution or i can be thought of as the middle 50% of the distribution. A point become an outlier if it is above the 75^{th} or below the 25^{th} percentile by a factor of 1.5 times the interquartile range.

rm.OutLiear_Type: Removal of rows where for “Air_temperature_K”, “Process_temperature_K”, “Rotational_speed_rpm”,“Torque_Nm” or "Tool_wear_min data points are out not with in the second or third quartile.

rm.OutLiear_Type: Removals of rows where for “Air_temperature_K”, “Process_temperature_K”, “Rotational_speed_rpm”,“Torque_Nm” or "Tool_wear_min data points are out not with in the second or third quartile when group by product type.

rm.OutLiear_Type: Removals of rows where for “Air_temperature_K”, “Process_temperature_K”, “Rotational_speed_rpm”,“Torque_Nm” or "Tool_wear_min data points are out not with in the second or third quartile when group target response column.

rm.OutLiear_Type: Removals of rows where for“Air_temperature_K”, “Process_temperature_K”,“Rotational_speed_rpm”,“Torque_Nm” or "Tool_wear_min data points are out not with in the second or third quartile when grouped by both product type andtarget response column.

Sampling:

Upsampling is a process that synthetically generates additional data points (corresponding to minority class) are added into the dataset. After doing so all labels in the data set have equal proportions, following this procedure the model from inclining towards the majority class.

Down sampling was considered at the early stage but initial testing showed that the resulting samples were to small to provided models enough data for models to have provide accurate results.

Class Weights:

Most machine learning algorithms are not especially useful when working with imbalanced data. Class weights can be used modify the current training algorithm to consider the imbalance of the data classes. This is dine by giving different weights for the class based on the proportions of class with in the training data sets. The weights will influence the classification of the classes during the training phase. The whole purpose is to penalize the misclassification made by the minority class by setting a higher class weight and at the same time reducing weight for the majority class. For this reason testing using Up sampling and case weight have been removed as up sampling create equal proportions of the classes, essentially making all the class weights equal.

Two Class Modelling Results

Ranger Current Results:

Show 15 **entries** **Search:**

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	AUC	Balanced Accuracy	F0.5	F1	F2
target	ranger	No-Sampling	No_Treatment	No_Weights	0.9752	0.7401	0.6957	0.5981	0.5246
target	ranger	up	No_Treatment	No_Weights	0.9699	0.8289	0.7097	0.6929	0.6769
target	ranger	No-Sampling	rm_OutLiear_No_Groups	No_Weights	0.9592	0.681	0.6897	0.5161	0.4124
target	ranger	up	rm_OutLiear_No_Groups	No_Weights	0.9208	0.8037	0.4585	0.5122	0.5801
target	ranger	No-Sampling	rm_OutLiear_Type	No_Weights	0.9628	0.7035	0.7105	0.5567	0.4576
target	ranger	up	rm_OutLiear_Type	No_Weights	0.9195	0.8126	0.4864	0.5375	0.6006
target	ranger	No-Sampling	rm_OutLiear_target	No_Weights	0.9431	0.7176	0.6776	0.5631	0.4817
target	ranger	up	rm_OutLiear_target	No_Weights	0.9214	0.7832	0.6552	0.623	0.5938
target	ranger	No-Sampling	rm_OutLiear_Type_target	No_Weights	0.9411	0.7179	0.6905	0.5686	0.4833
target	ranger	up	rm_OutLiear_Type_target	No_Weights	0.9253	0.7832	0.6552	0.623	0.5938
target	ranger	No-Sampling	No_Treatment	Applied	0.9739	0.8415	0.6425	0.6619	0.6825
target	ranger	No-Sampling	rm_OutLiear_No_Groups	Applied	0.9415	0.7398	0.6838	0.5926	0.5229
target	ranger	No-Sampling	rm_OutLiear_Type	Applied	0.9408	0.7393	0.6612	0.5818	0.5195
target	ranger	No-Sampling	rm_OutLiear_target	Applied	0.9134	0.7681	0.6383	0.6	0.566

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	AUC	Balanced Accuracy	F0.5	F1	F2
target	ranger	No-Sampling	rm_OutLiear_Type_target	Applied	0.9075	0.7759	0.656	0.6167	0.5818

Showing 1 to 15 of 15 entries Previous 1 Next

XgbTree Current Results:

Show 10 **entries** **Search:**

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	AUC	Balanced Accuracy	F0.5	F1	F2
target	xgbTree	No-Sampling	No_Treatment	No_Weights	0.9768	0.7239	0.6303	0.5505	0.4886
target	xgbTree	up	No_Treatment	No_Weights	0.9771	0.9011	0.665	0.7152	0.7736
target	xgbTree	No-Sampling	rm_OutLiear_No_Groups	No_Weights	0.9441	0.7328	0.6982	0.5905	0.5116
target	xgbTree	up	rm_OutLiear_No_Groups	No_Weights	0.9554	0.8104	0.5932	0.6087	0.625
target	xgbTree	No-Sampling	rm_OutLiear_Type	No_Weights	0.9508	0.7101	0.6667	0.549	0.4667
target	xgbTree	up	rm_OutLiear_Type	No_Weights	0.9669	0.8256	0.6077	0.6286	0.6509
target	xgbTree	No-Sampling	rm_OutLiear_target	No_Weights	0.9419	0.6933	0.5752	0.4906	0.4276
target	xgbTree	up	rm_OutLiear_target	No_Weights	0.9452	0.8329	0.6081	0.6338	0.6618
target	xgbTree	No-Sampling	rm_OutLiear_Type_target	No_Weights	0.943	0.7398	0.6838	0.5926	0.5229
target	xgbTree	up	rm_OutLiear_Type_target	No_Weights	0.9535	0.8632	0.6347	0.6712	0.7122

Showing 1 to 10 of 10 entries Previous 1 Next

Multiclass Model Test Case Results

Ranger Current Results:

Show 15 **entries** **Search:**

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	AUC	Kappa	LogLoss	Mean_Erro
Failure_Type	ranger	No-Sampling	No_Treatment	No_Weights	0.9084	0.5297	0.0832	0.008
Failure_Type	ranger	up	No_Treatment	No_Weights	0.8958	0.4781	0.13	0.008
Failure_Type	ranger	No-Sampling	rm_OutLiear_No_Groups	No_Weights	0.8916	0.4411	0.1031	0.008
Failure_Type	ranger	up	rm_OutLiear_No_Groups	No_Weights	0.8894	0.2925	0.1852	0.009
Failure_Type	ranger	No-Sampling	rm_OutLiear_Type	No_Weights	0.8929	0.4248	0.1009	0.00
Failure_Type	ranger	up	rm_OutLiear_Type	No_Weights	0.8721	0.2929	0.1988	0.010
Failure_Type	ranger	No-Sampling	rm_OutLiear_target	No_Weights	0.9212	0.4413	0.1094	0.01
Failure_Type	ranger	up	rm_OutLiear_target	No_Weights	0.8669	0.4083	0.1742	0.01
Failure_Type	ranger	No-Sampling	rm_OutLiear_Type_target	No_Weights	0.8998	0.4302	0.1113	0.0
Failure_Type	ranger	up	rm_OutLiear_Type_target	No_Weights	0.8787	0.4221	0.1567	0.010
Failure_Type	ranger	No-Sampling	No_Treatment	Applied	0.8924	0.6003	0.1642	0.009
Failure_Type	ranger	No-Sampling	rm_OutLiear_No_Groups	Applied	0.8814	0.4965	0.2128	0.01
Failure_Type	ranger	No-Sampling	rm_OutLiear_Type	Applied	0.8696	0.5265	0.2267	0.010
Failure_Type	ranger	No-Sampling	rm_OutLiear_target	Applied	0.8965	0.5614	0.1737	0.01
Failure_Type	ranger	No-Sampling	rm_OutLiear_Type_target	Applied	0.869	0.5674	0.1858	0.010

Showing 1 to 15 of 15 entries Previous 1 Next

xgbTree Current Results:

Show 10 **entries** **Search:**

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	AUC	Kappa	LogLoss	Mean_Erro
Failure_Type	xgbTree	No-Sampling	No_Treatment	No_Weights	0.9209	0.6288	0.0751	0.007
Failure_Type	xgbTree	up	No_Treatment	No_Weights	0.8611	0.6451	0.096	0.008
Failure_Type	xgbTree	No-Sampling	rm_OutLiear_No_Groups	No_Weights	0.9016	0.5173	0.0957	0.008
Failure_Type	xgbTree	up	rm_OutLiear_No_Groups	No_Weights	0.8838	0.5238	0.1093	0.010
Failure_Type	xgbTree	No-Sampling	rm_OutLiear_Type	No_Weights	0.8971	0.4976	0.0956	0.008
Failure_Type	xgbTree	up	rm_OutLiear_Type	No_Weights	0.85	0.597	0.1041	0.00
Failure_Type	xgbTree	No-Sampling	rm_OutLiear_target	No_Weights	0.9102	0.3829	0.1246	0.018
Failure_Type	xgbTree	up	rm_OutLiear_target	No_Weights	0.8713	0.5409	0.1451	0.012
Failure_Type	xgbTree	No-Sampling	rm_OutLiear_Type_target	No_Weights	0.9365	0.4173	0.1052	0.015
Failure_Type	xgbTree	up	rm_OutLiear_Type_target	No_Weights	0.8603	0.5108	0.1387	0.013

Showing 1 to 10 of 10 entries

Previous 1 Next

3.4 Model Analysis

2 Class Models down selection

Having run the baselines and model development options its time to down select modelling options before further tuning attempts, for model down selection in have choose 4 metrics to help with down selection these are: F0.5 - fbeta 0.5, AUC, RSME & balanced Acc. I shall select the top 2 performers for each metric, remove any duplicate models that are selected more than once, and then tune to these to help down select the final model.

```
## [1] 8
```

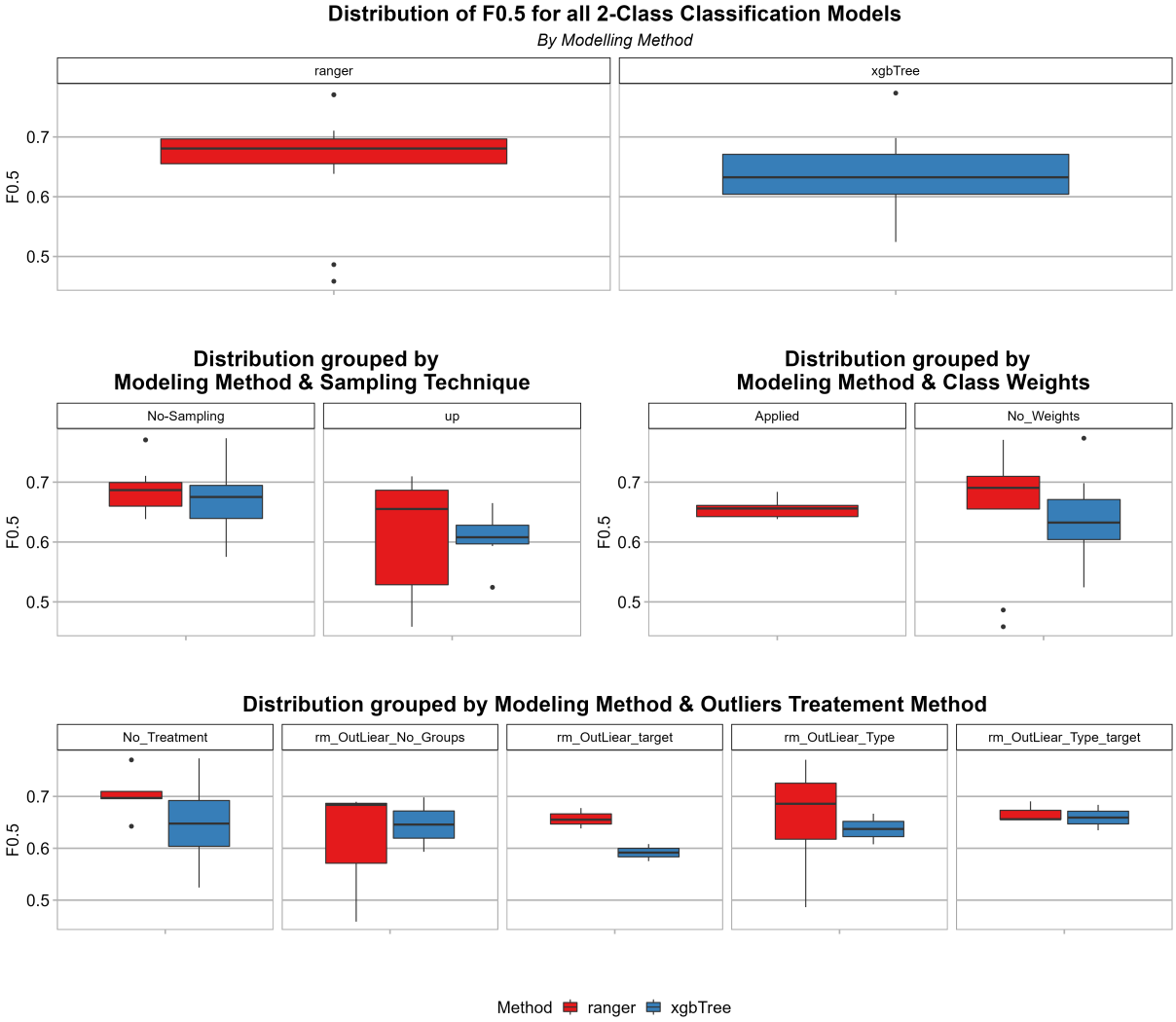
Models Selected for tuning

	Method	target	Sampling_Method	OutLiear_Treatment	Class_Weights
9	ranger	target	No-Sampling	rm_OutLiear_Type	No_Weights
3	ranger	target	up	No_Treatment	No_Weights
4	xgbTree	target	up	No_Treatment	No_Weights
1	ranger	target	No-Sampling	No_Treatment	No_Weights
2	xgbTree	target	No-Sampling	No_Treatment	No_Weights

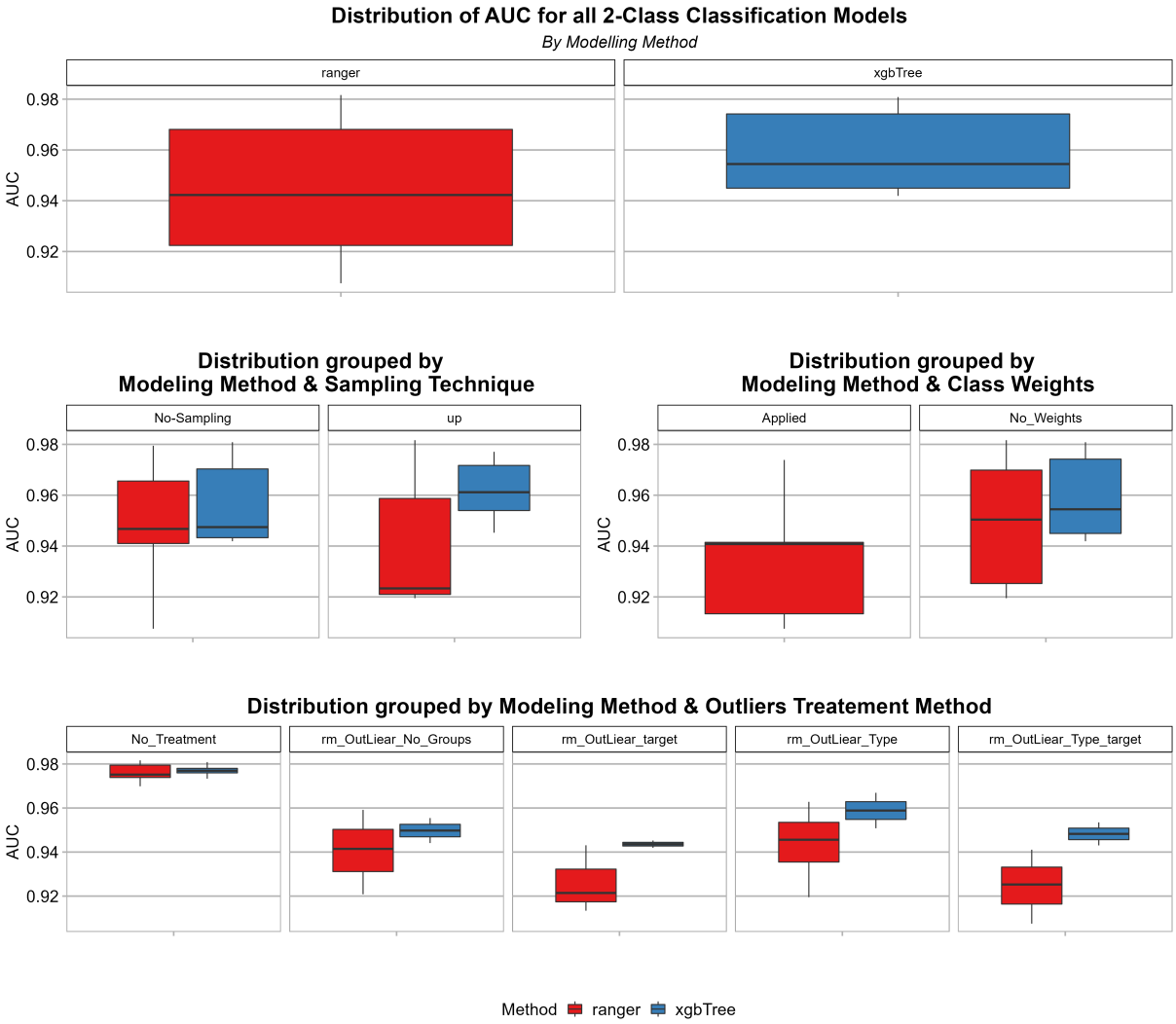
Best performing Two Class Models Results with Tuning

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	F0.5	RSME	Overall_Error	AUC	F1	Balanced Accuracy	Accuracy
target	ranger	No-Sampling	rm_OutLiear_Type	No_Weights	0.7706767	0.1304167	0.0170085	0.9503990	0.7068966	0.8082781	0.9829915
target	ranger	up	No_Treatment	No_Weights	0.6969697	0.1414567	0.0200100	0.9816348	0.6969697	0.8433115	0.9799900
target	xgbTree	up	No_Treatment	No_Weights	0.5243446	0.1884616	0.0355178	0.9733026	0.6120219	0.9084638	0.9644822
target	ranger	No-Sampling	No_Treatment	No_Weights	0.7704403	0.1245301	0.0155078	0.9794518	0.7596899	0.8675908	0.9844922
target	xgbTree	No-Sampling	No_Treatment	No_Weights	0.7733813	0.1284845	0.0165083	0.9808431	0.7226891	0.8231709	0.9834917

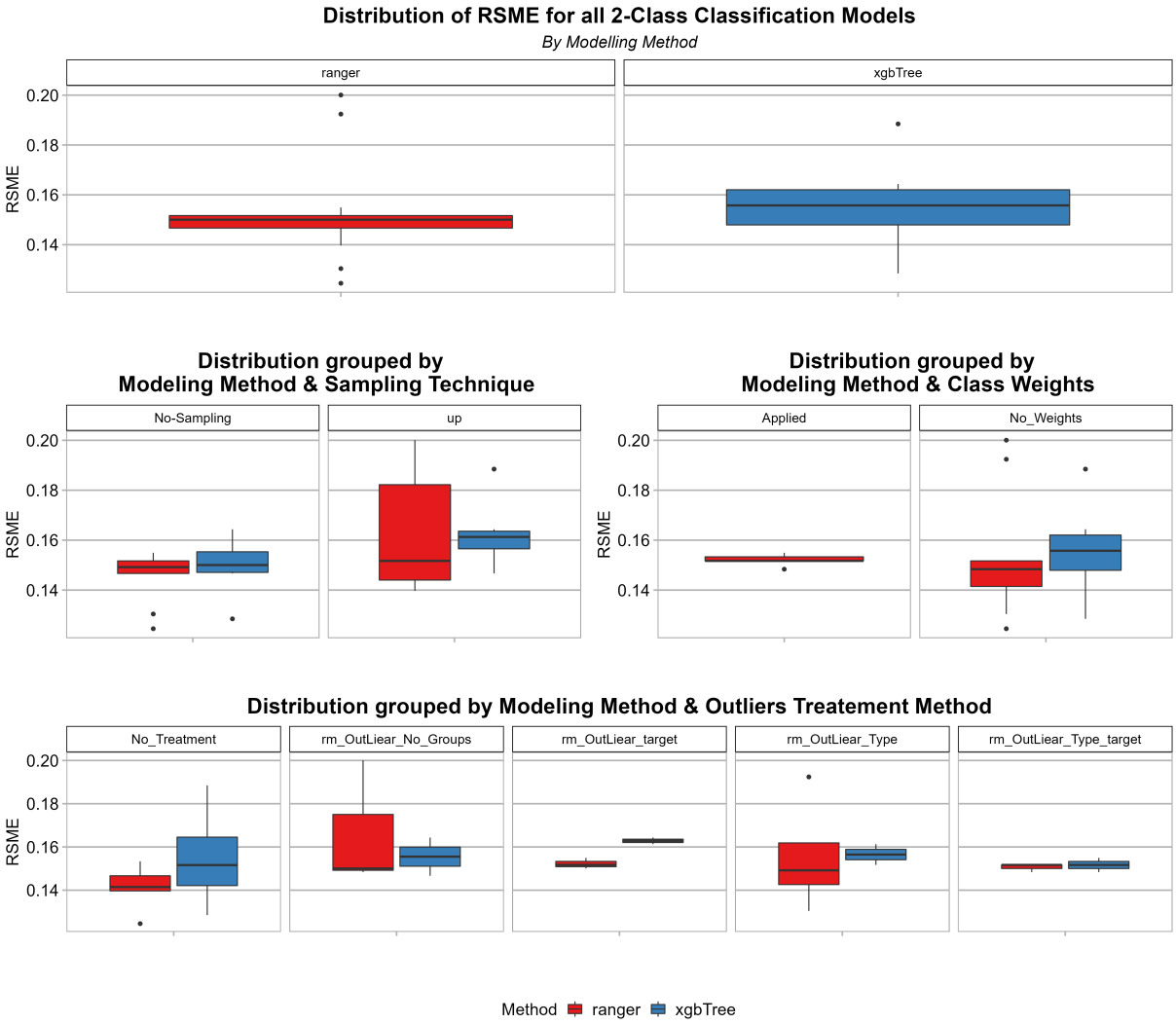
2 Class Modelling Plots & Review



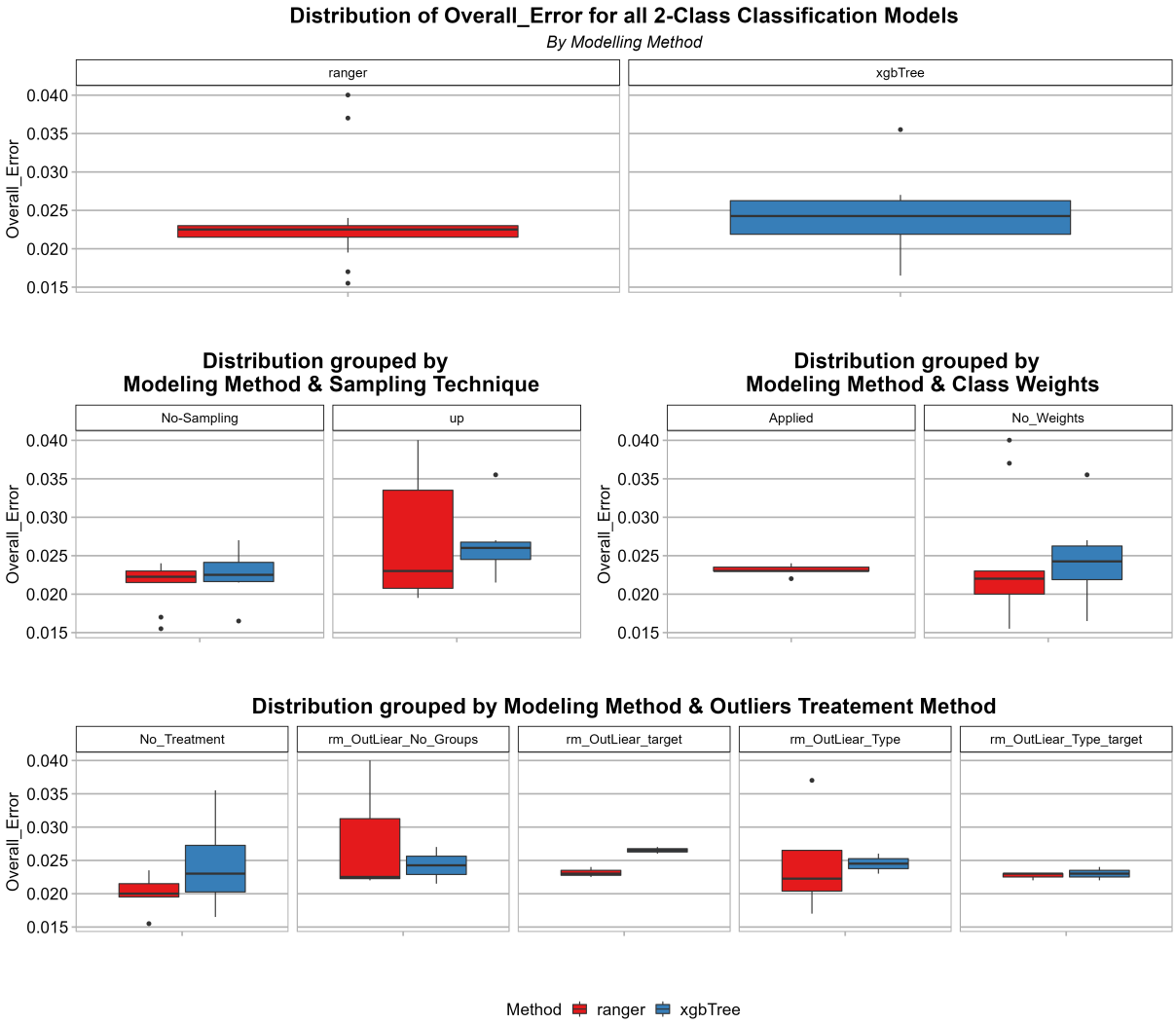
For F0.5 xgbTree is the performs better that than ranger, having a high mean for F0.5 across all techniques applied. UP sampling has an adverse effect on both modelling methods, resulting in a considerable drop in F0.5. Applying class weight to ranger created a slight shift to F0.5 by increasing the mean F0.5 score when compared with having no weights applied, but ranger achieved higher F0.5 scores when weights were not applied. Any improvement achieved by applying class weight to ranger minimal if any. Removing outliers with no groupings by target or type and removing outliers when grouping by Type both does seems to lift the F0.5 score that the trained models achieved. xgbTree gains the most improvement of the two modelling techniques. It worth noting the Removing outliers when grouping the data by type and target resulted in higher F5.0 scores for ranger, xgbTree did not respond as positively to the same outlier removal technique.



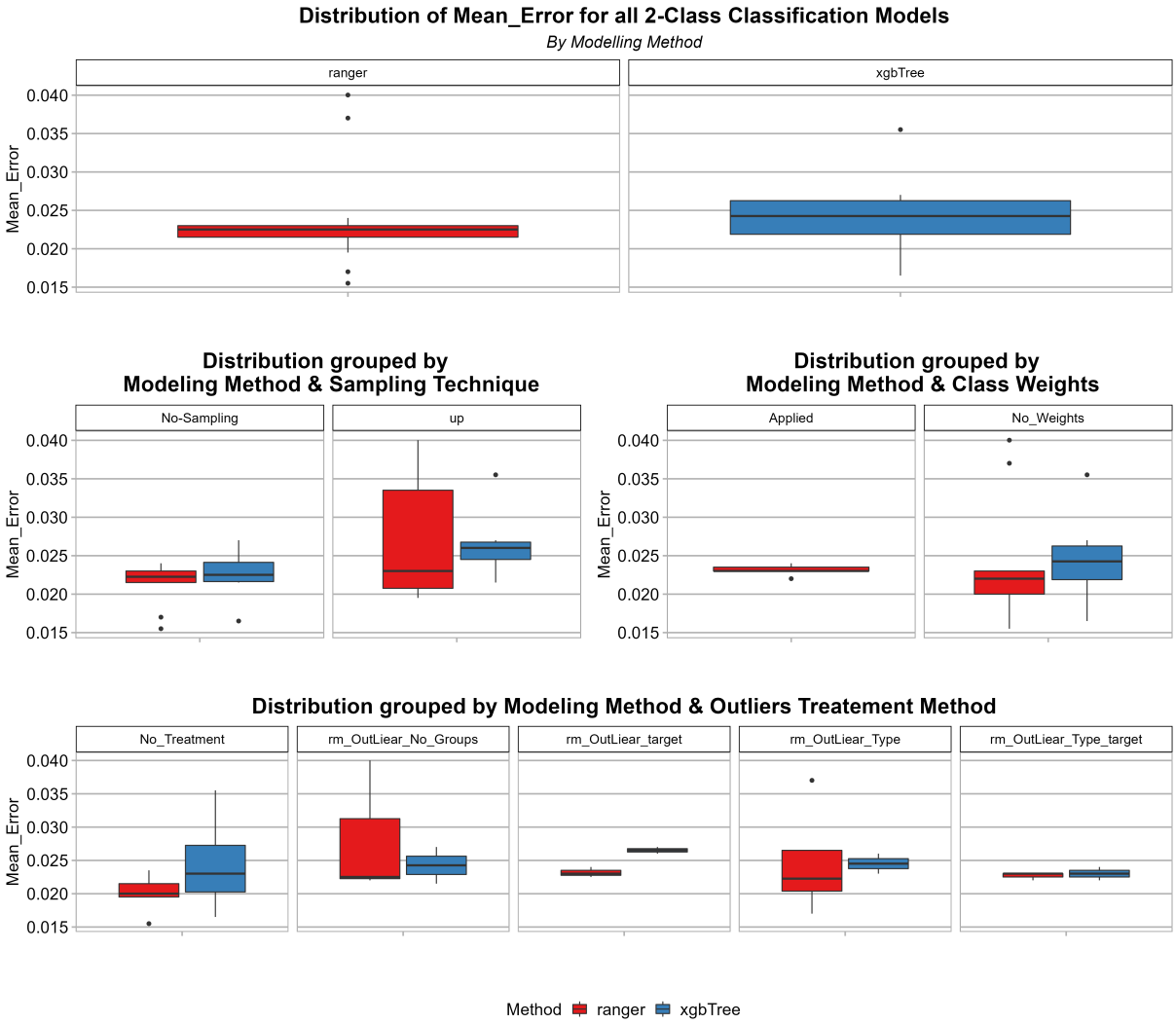
XgbTree definelyperforms better than ranger achieving higher AUC score across the board. The techniques for outlier removal did not improve the overall scores the models achieved.



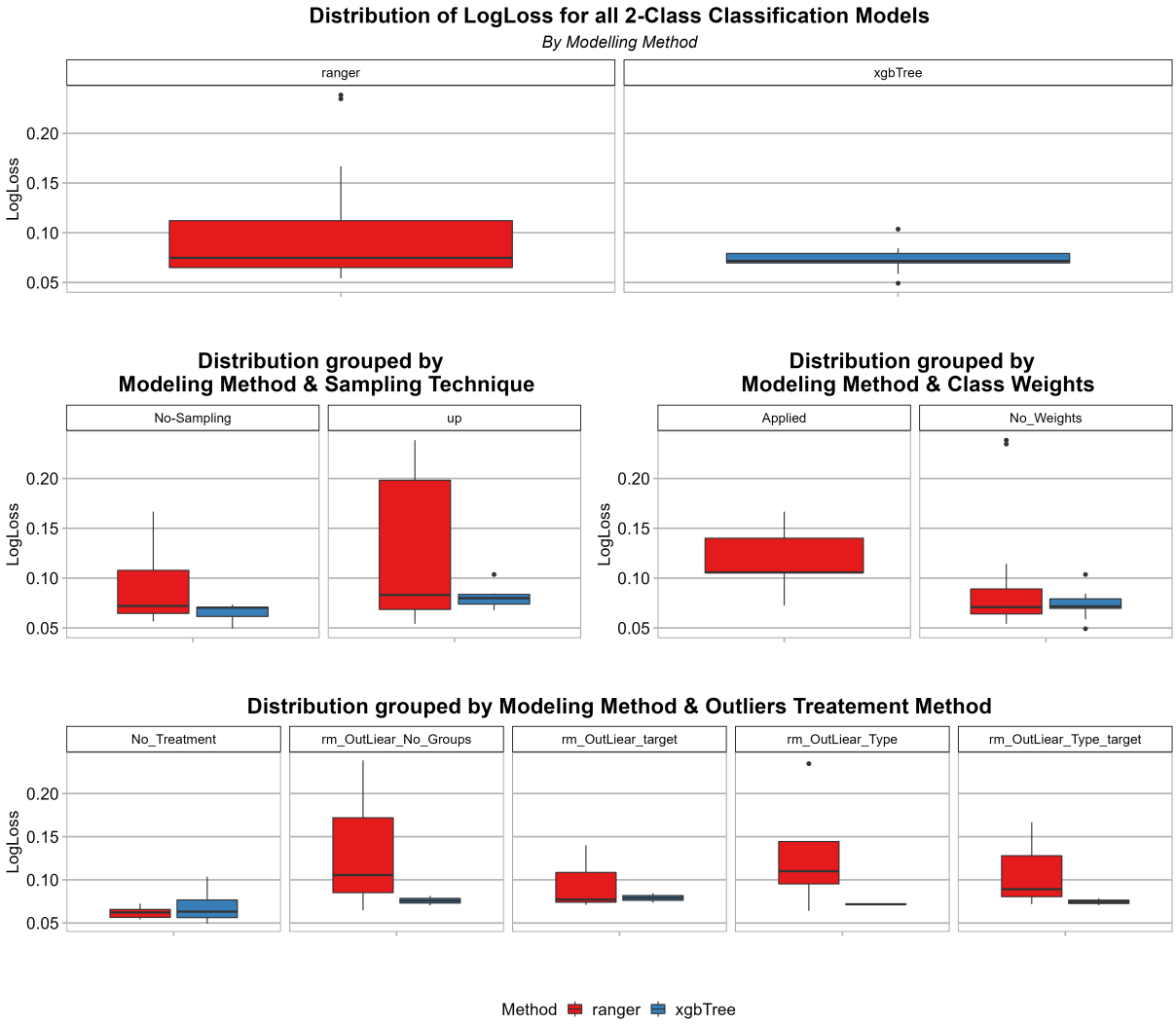
Interestingly Removing outliers' grouped by type had a positive effect in reducing the RSME for xgbTree and negligible effect on ranger, but for removing outliers when group by type and target has a positive effect on ranger but hardly any effect on xgbTree.



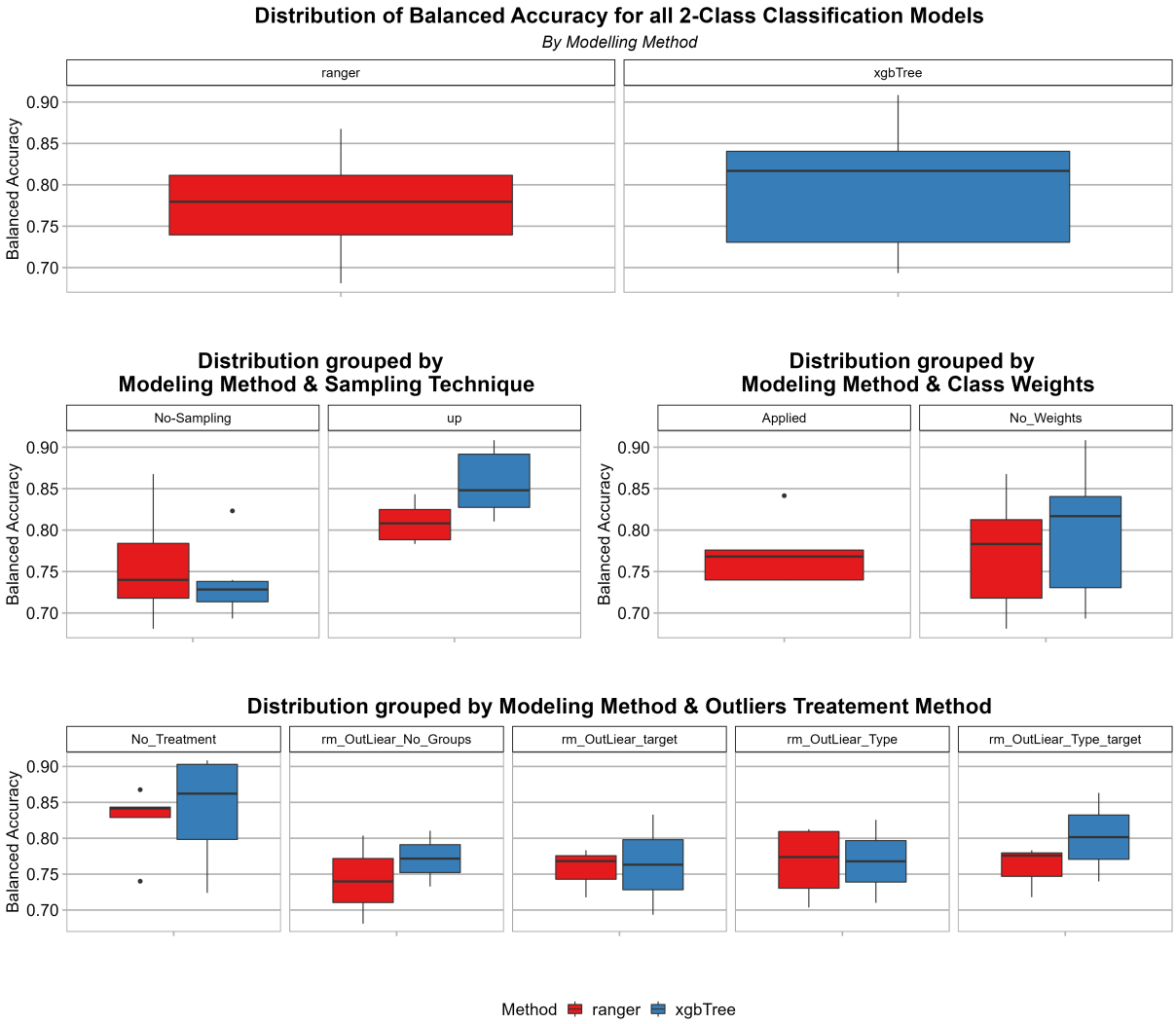
Up sampling had a detrimental effect on the overall error of the models, removal of outliers with no grouping does seem to reduce the overall error of the models.



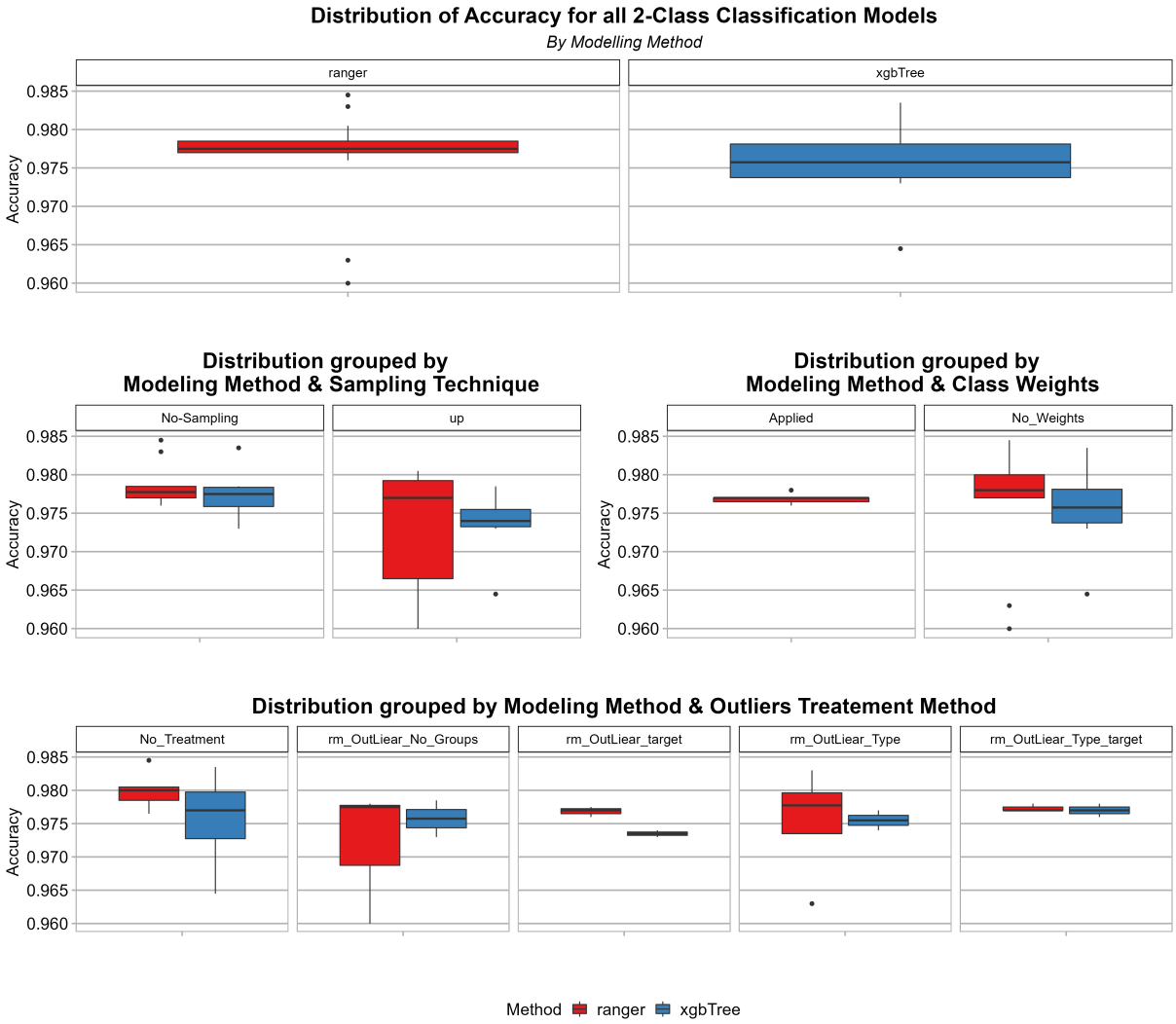
Up sampling does not improve the mean error of the models, removing outliers does seem to reduce the sizes of the distributions for both model techniques and does reduce the average mean error.



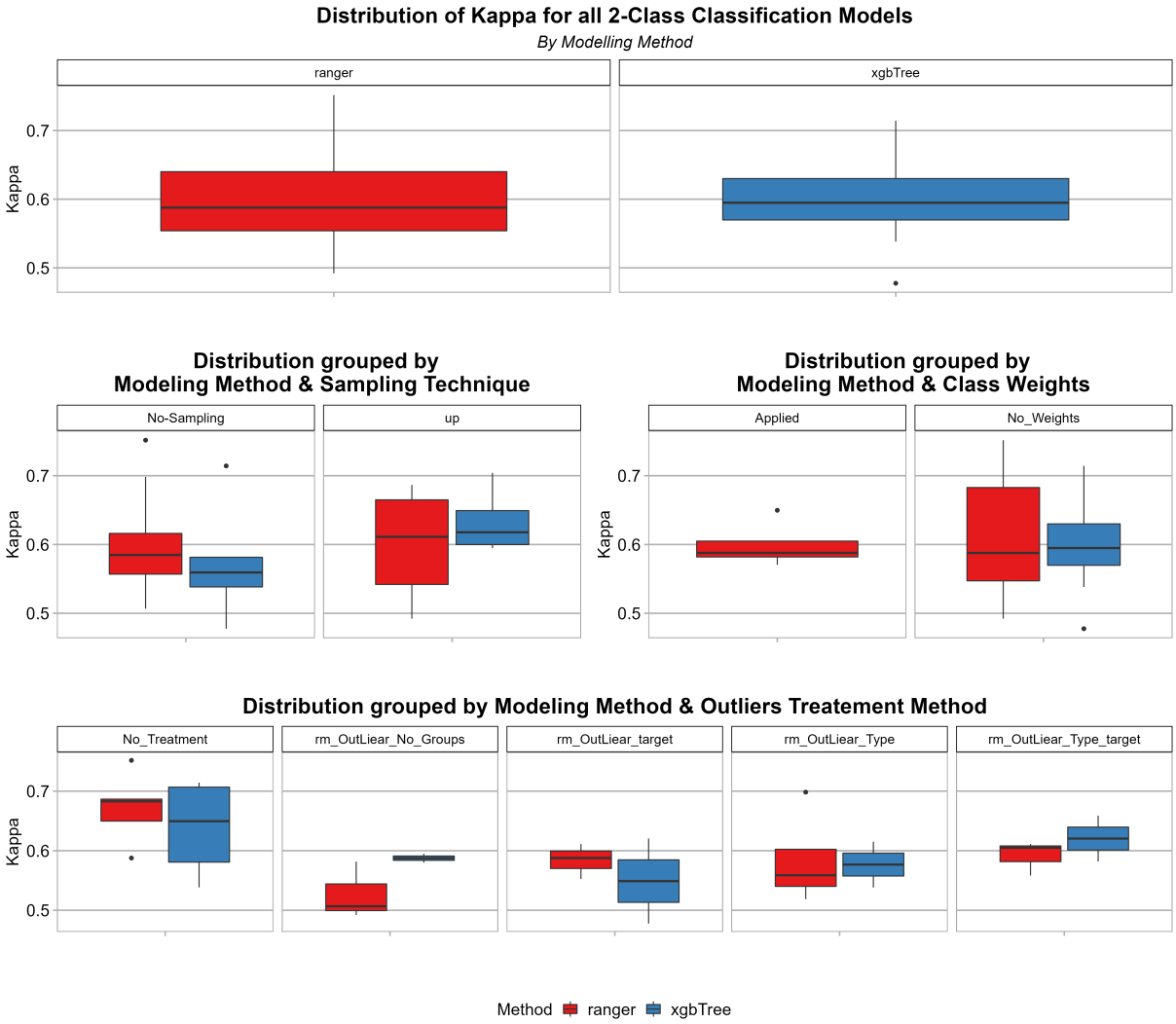
xgbTree performs much better than ranger across all techniques. Interestingly the two-modelling method performed very differently based on this metric, it is clear to see here that xgbTree performs much better than ranger in reducing LogLoss during the modelling process.



Interesting to see how up sampling allowed xgbTree to achieve a 7%-8% improvement in balancedaccuracy while ranger suffers 7% decrease in it balanced accuracy when up sampling is used. Ranger also suffered a considerable drop when outliers where removed when group the data by type.



It possible to see how removal of outliers does provide improvement here by the reduced size of thebox plot distributions.



Removing outliers by type again can be shown to have a positive effect on xgbTree method and a detrimental affect on ranger, allowing xgbTree to achieved some of it highest Kappa scores while ranger achieved its lowest Kappa scores for the same outlier treatment method.

Multi Class Models Down selection

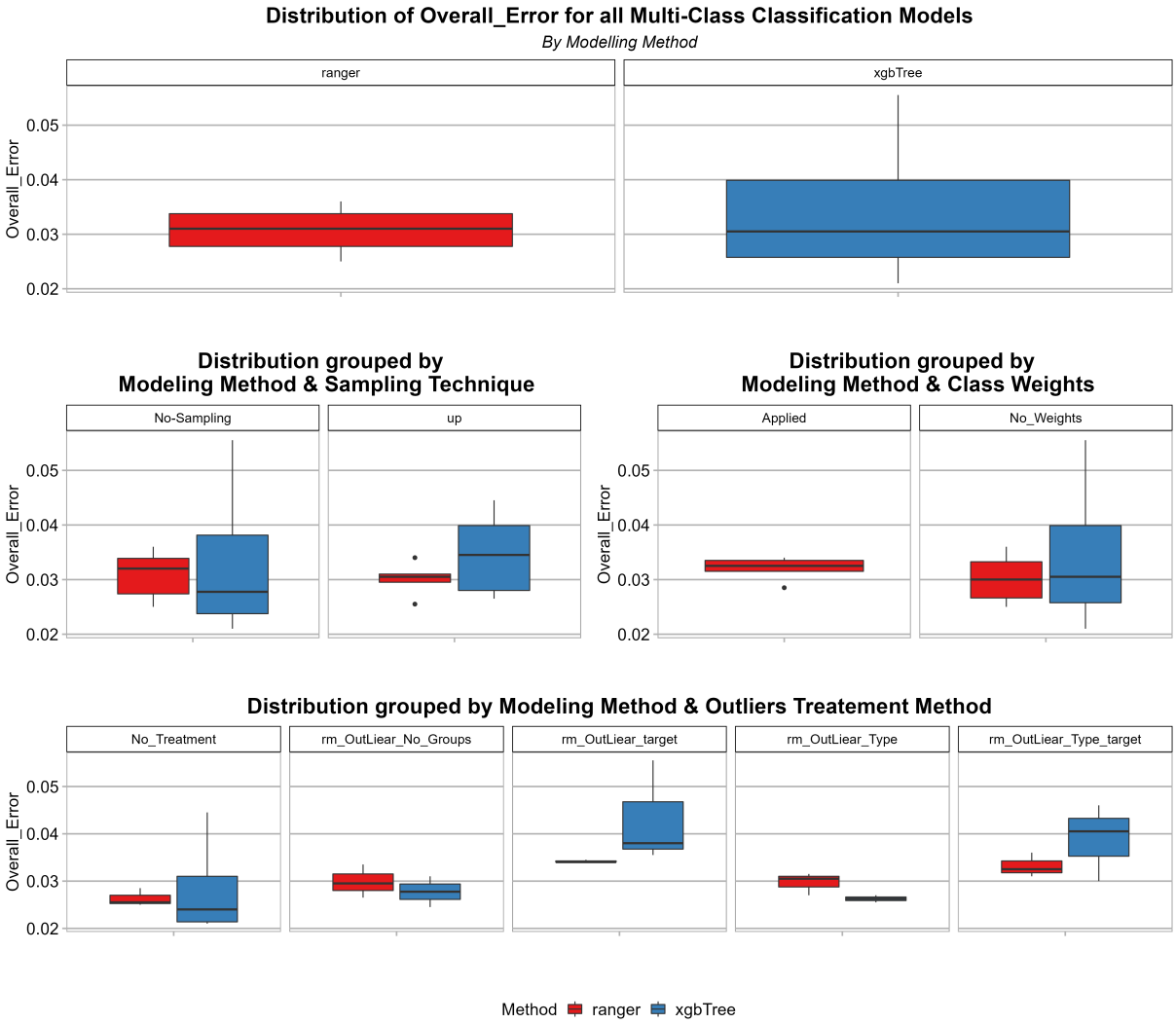
Mutli-Class Classification Models Selected for tuning

Method	target	Sampling_Method	OutLiear_Treatment	Class_Weights
xgbTree	Failure_Type	No-Sampling	rm_OutLiear_target	No_Weights
xgbTree	Failure_Type	No-Sampling	rm_OutLiear_Type_target	No_Weights
xgbTree	Failure_Type	up	No_Treatment	No_Weights
xgbTree	Failure_Type	No-Sampling	No_Treatment	No_Weights

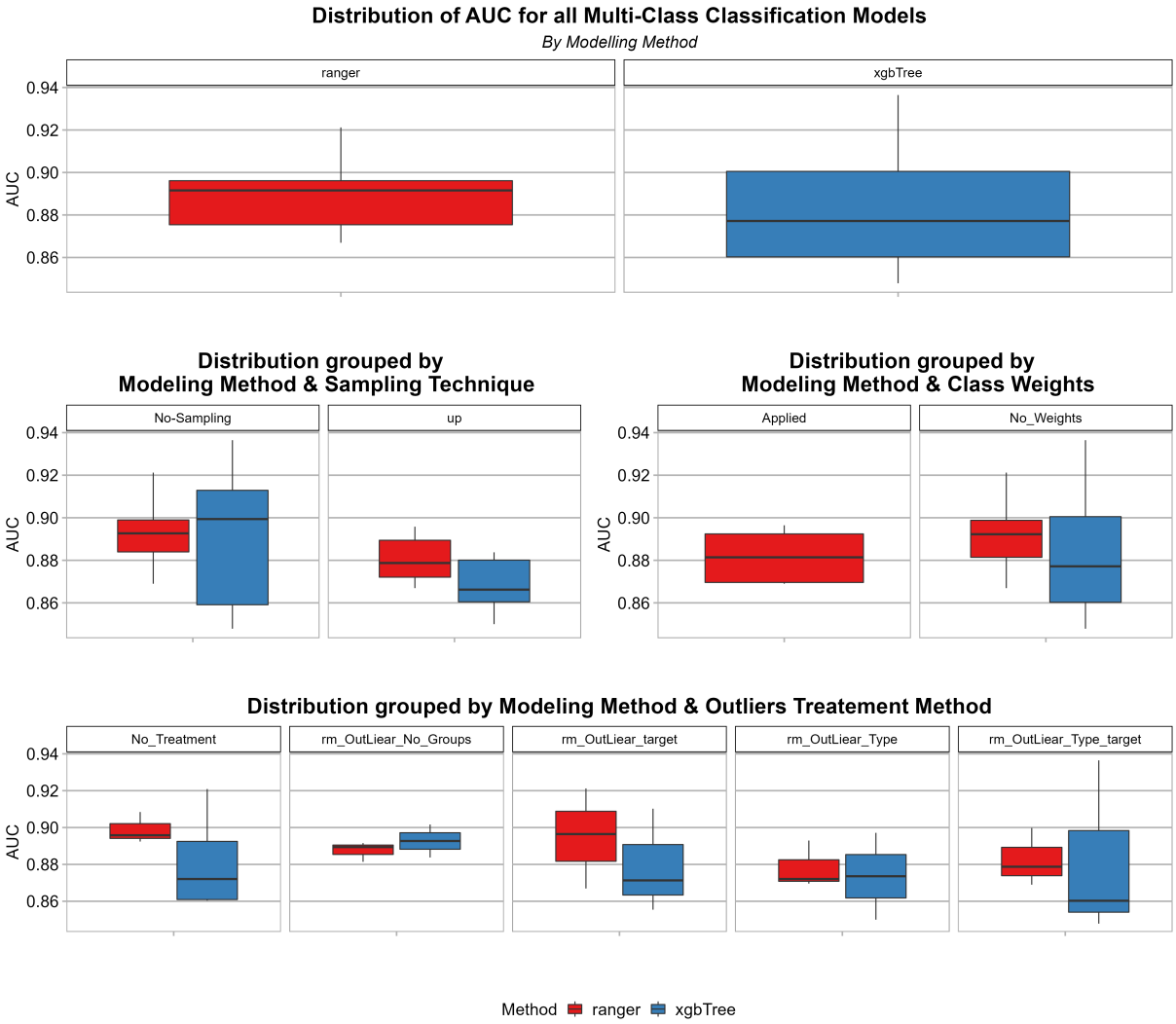
Best performing Mutli-Class Classification Models Results with Tuning

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights	Overall_Error	AUC	Mean_Error	LogLoss	Accuracy	Kappa
Failure_Type	xgbTree	No-Sampling	rm_OutLiear_target	No_Weights	0.0355178	0.8554311	0.0118393	0.1320054	0.9644822	0.5213413
Failure_Type	xgbTree	No-Sampling	rm_OutLiear_Type_target	No_Weights	0.0300150	0.8478538	0.0100050	0.1280851	0.9699850	0.5646524
Failure_Type	xgbTree	up	No_Treatment	No_Weights	0.0445223	0.8830160	0.0148408	0.1465425	0.9554777	0.5257679
Failure_Type	xgbTree	No-Sampling	No_Treatment	No_Weights	0.0210105	0.8603443	0.0070035	0.0745872	0.9789895	0.6462215

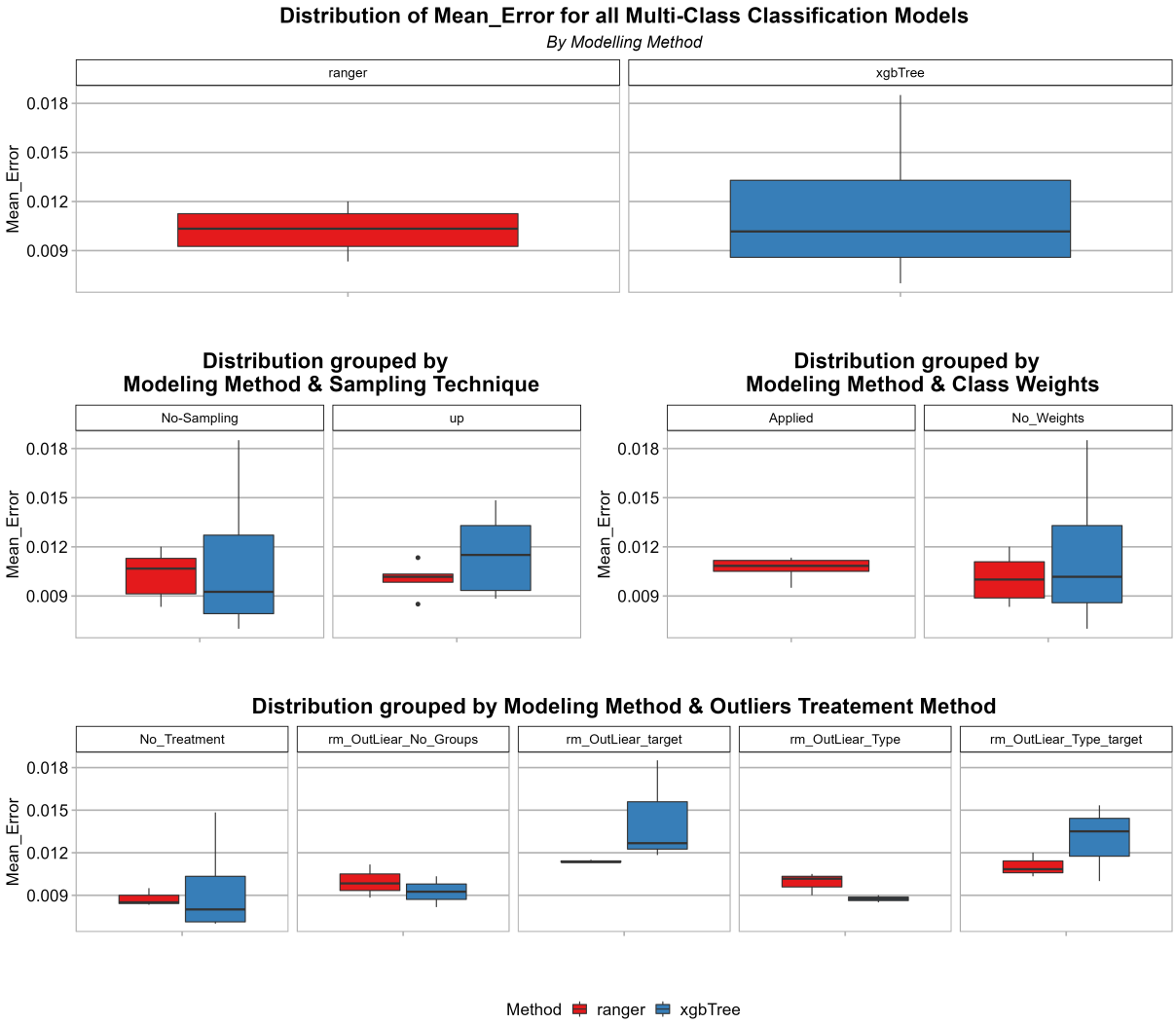
Mutli Class Modelling Plots & Review



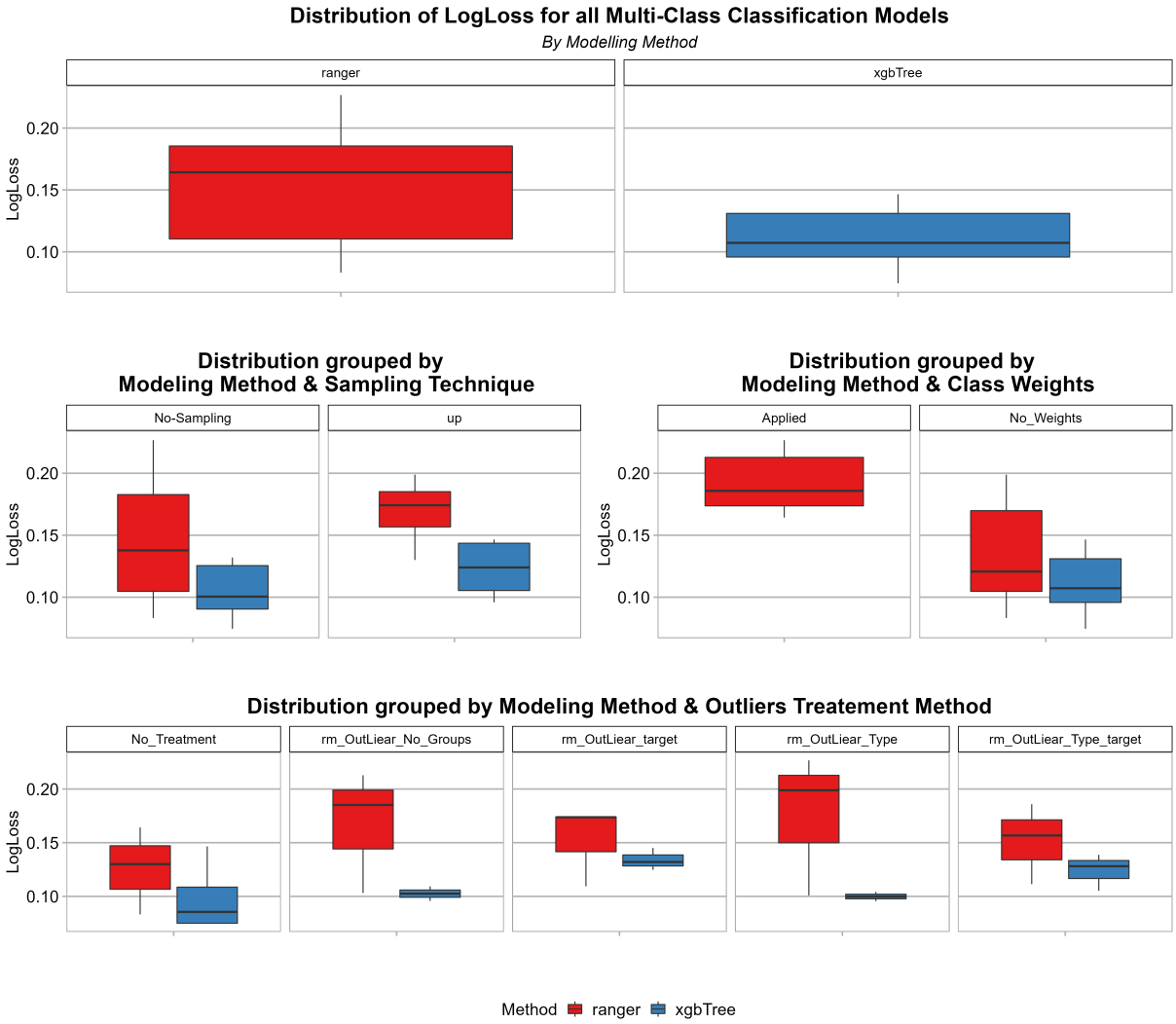
The overall Error does not see any improve from any of the techniques applied, and generates the best performance measures when no sampling or outliers removal techniques are applied.



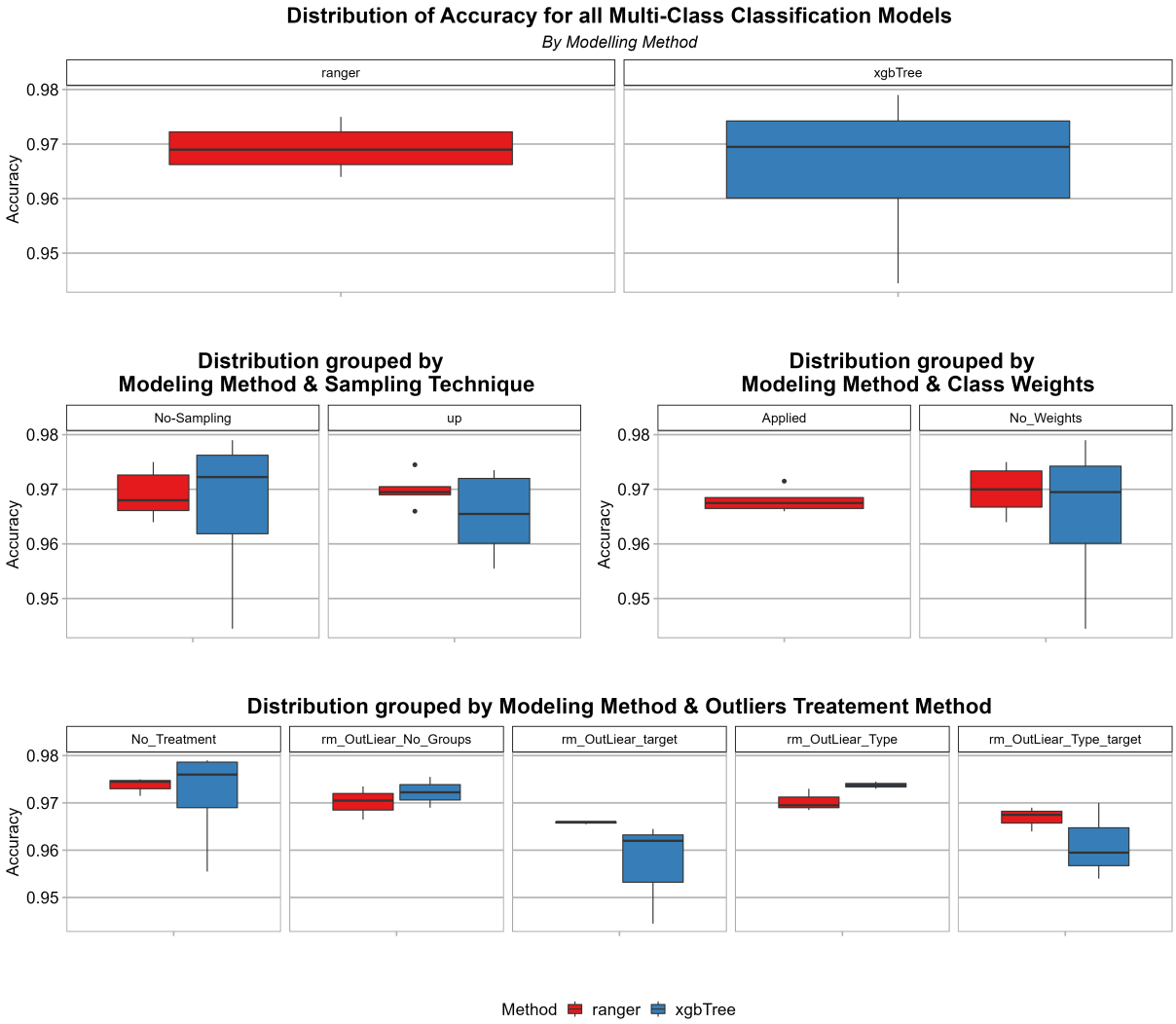
The AUC measure does not seem to improve from any of the techniques that have been applied here. Ranger has larger distribution of the values it achieved as well as higher and lower values than xgbTree.



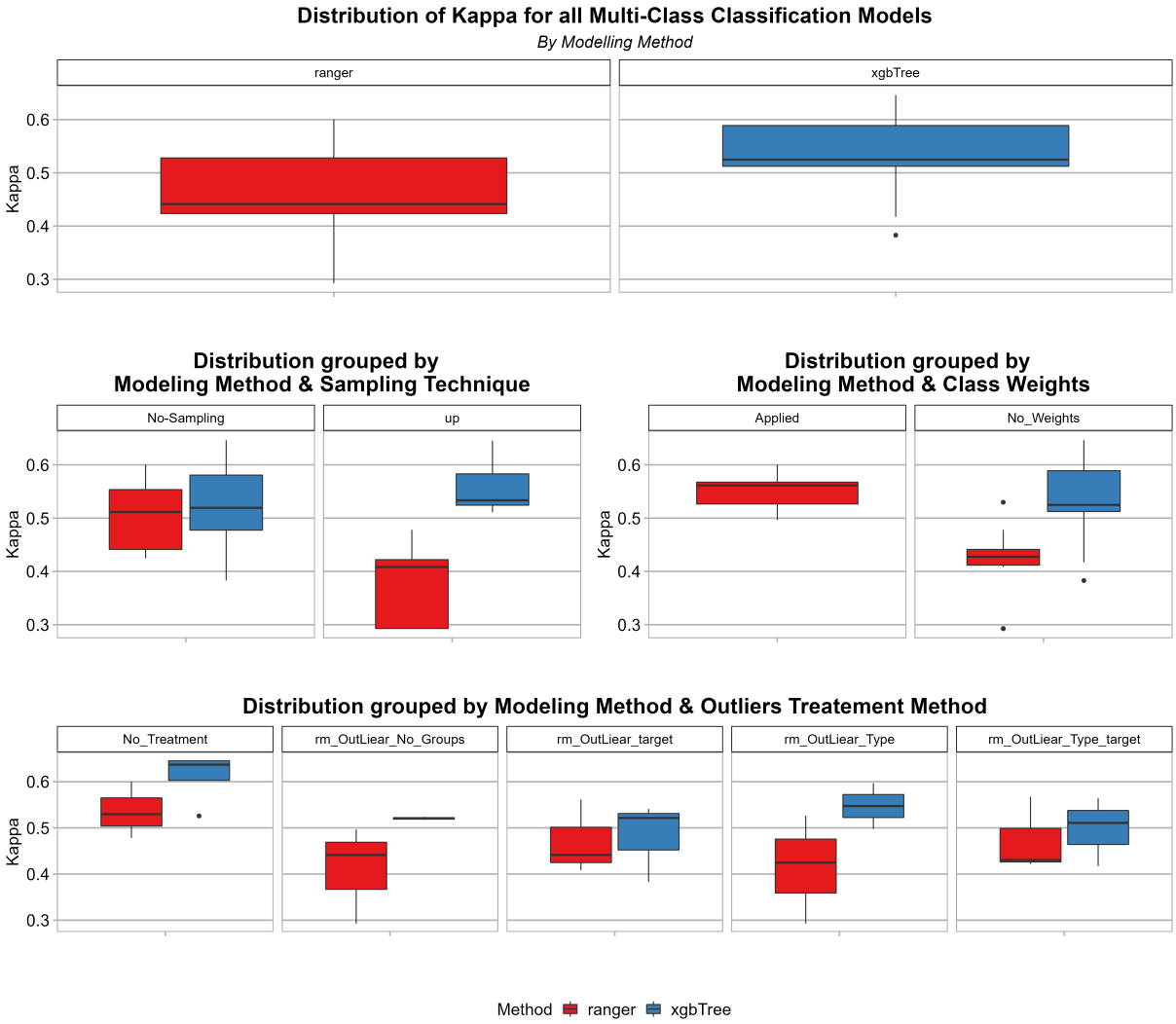
The mean error does not see any improve from any of the techniques applied, and generates the best performance measures when no sampling or outliers removal techniques are applied.



Overall xgbTree does seem to perform better than ranger for this metric, though both model method seem to be negatively impact by the various outlier removal techniques it seems that the detrimental impact to ranger is much more pronounced than theimpact for xgbTree.



The mean error does not see any improvement from any of the techniques applied, and generates the best performance measures when no sampling or outliers removal techniques are applied.



It appears that Up sampling had a negative impact for ranger and xgbTree but the affect is much more pronounced for ranger than xgbTree.

3.5 Final Model Selection

Two Class Classification: Data Manipulation Parameters selected for Validation Model

Model Selected for Two Class Classification Validation

	target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights
27	target	ranger	up	No_Treatment	No_Weights

Multi Class Classification: Data Manipulation Parameters selected for Validation Model

Model Selected for Multi Class Classification Validation

	target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights
18	Failure_Type	xgbTree	No-Sampling	rm_OutLiear_Type_target	No_Weights

Two Class Model Results

Modelling varibles that were down selected for Final Vaildation

target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights
target	ranger	up	No_Treatment	No_Weights

Comparison of Final Models Results to Model Baseline

	Baselines	Final Vaildation	% Increase on Baselines	Delta between Final & Baseline
AUC	0.96987	0.98874	1.94532	0.01887
Balanced Accuracy	0.82894	0.86334	4.15067	0.03441
F0.5	0.70968	0.81940	15.46063	0.10972
F1	0.69291	0.78400	13.14545	0.09109
F2	0.67692	0.75153	11.02203	0.07461
Kappa	0.68285	0.77707	13.79669	0.09421
LogLoss	0.06568	0.04252	35.26601	0.02316
Mean_Error	0.01951	0.01351	30.76923	0.00600

	Baselines	Final Vaildation	% Increase on Baselines	Delta between Final & Baseline
RSME	0.13968	0.11622	16.79497	0.02346

Two Class Final COnfusion Matrix

	Fail	Pass
Fail	49	9
Pass	18	1923

Even though the final modelling parameters that were selected for the validation were the same as the baseline model meaning that not up-sampling, no outliers treatment methods, or class weights were applied the final validation model performed well and resulted in strong improvements on the baseline statistics. The AUC values increase the least percentage wise from 0.96987 to 0.98874 which is a 1.94532% improvement, even though the improvement achieved was not huge still represents and model that is able to predict well between the two class. The final achieved balanced accuracy is a reasonable result and represents a significant improvement on the baselines. The final two class model do produce satisfactory results and all measures do demonstrate a model that does have good ability to classify process failures.

Modelling variables that were down selected for Final Validation

	target	Method	Sampling_Method	OutLiear_Treatment	Class_Weights
Conf$overall	Failure_Type	xgbTree	No-Sampling	rm_OutLiear_Type_target	No_Weights

Comparison of Final Models Results to Model Baseline

	Baselines	Final Vaildation	% Increase on Baselines	Difference between Final & Baseline
AUC	0.93645	0.89092	4.86234	0.04553
Mean_Error	0.01534	0.00834	45.65217	0.00700
LogLoss	0.10516	0.09656	8.18310	0.00861
Accuracy	0.95398	0.97499	2.20241	0.02101
Kappa	0.41730	0.66295	58.86604	0.24565

Final Mutli-Class Classification Confusion Matrix

	Heat.Dissipation.Failure	No.Failure	Overstrain.Failure	Power.Failure	Random.Failures	Tool.Wear.Failure
Heat.Dissipation.Failure	21	1	0	0	0	0
No.Failure	1	1899	6	1	4	7
Overstrain.Failure	1	0	10	0	0	0
Power.Failure	0	25	0	18	0	1
Random.Failures	0	0	0	0	0	0
Tool.Wear.Failure	0	3	0	0	0	1

Final modelling parameters that were selected for the validation included up-sampling but no outliers treatment methods, or class weights were applied. The final validation model performed well with and resulted in modest improvements on the baseline statistics. the final model showed good abilities to predict Heat Dissipation Failure, overstrain failures, and power failures. The models struggled to predict Tool failure accurately and failed to predict any of the random failure at all. Overall, it is felt that the final model does have some good strengths, even though the final model only predicted one tool wear failures correctly, it is felt that this is still a good achievement considering the class imbalances of the data set. Although the models do have represent an improvement on baselines, there is opportunity to further improve the predictive power of this model to reduce the number of false predictions. If the models at least predicted a random failure even if the prediction were wrong feels like it would also strengthen the model as right now the models fail to predict any random failures at all. The absence of random failure predictions definitely feels like a weakness for this model, and would be a area that worth focusing on in future models.

4.1 Conclusion

The fact the two-class prediction model baseline and the final model used the same data is a little surprising, although random forest models do generally perform well with outliers it was not anticipated that after all the data modifications applied to the raw data that the baselines would be the one to be selected for the final model. That aside the final two class models is a solid model that does classify fails well and does perform well especially for AUC and Logloss.

The mutli-class Classification model does predict some cases well, as does improve on the initially baselines. It does ability to make a random failure prediction and does not predict tool wear accurately. That side the models does show strengths in predicting Heat Dissipation, overstrain, and power failures.

4.2 Future work and considerations

The following points might be worth considering in the future to further improve the Machine Failure classification engine:

- Dimensional reduction techniques being applied to the data and then applying some clustering technique evaluate any opportunities for those models to be used either standalone or as part of an ensemble to help strengthen the current models
- Treating tool wear as a ordinal data set to see if that improves on the classification engine ability to predict tool wear, or reduce the number of false predictions.

4.3 References

- Irizarry, Rafael A., “Introduction to Data Science: Data Analysis and Prediction Algorithms in R”, webpage:https://rafalab.github.io/dsbook/ (https://rafalab.github.io/dsbook/)
- LaTeX Equation Builder, webpage:https://latex.codecogs.com/eqneditor/editor.php (https://latex.codecogs.com/eqneditor/editor.php)
- knitr::kable and kableExtra Tutorial, webpage:https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html (https://cran.r-project.org/web/packages/kableExtra/vignettes/awesome_table_in_html.html)

- [Chapter 11 Random Forests | Hands-On Machine Learning with R (bradleyboehmke.github.io)] webpage: https://bradleyboehmke.github.io/HOML/random-forest.html#out-of-the-box-performance (https://bradleyboehmke.github.io/HOML/random-forest.html#out-of-the-box-performance)
- Handling Class Imbalance with R and Caret, webpage:[Handling Class Imbalance with R and Caret - An Introduction | Wicked Good Data dpmartin42.github.io)] webpage:https://dpmartin42.github.io/posts/r/imbalanced-classes-part-1 (https://dpmartin42.github.io/posts/r/imbalanced-classes-part-1)
- How to Remove Outliers in R, webpage:[How to Remove Outliers in R | R-bloggers](webpage:https://www.r-bloggers.com/2020/01/how-to-remove-outliers-in-r/ (https://www.r-bloggers.com/2020/01/how-to-remove-outliers-in-r/)
- Handling Imbalanced Data – Machine Learning, Computer Vision and NLP, webpage:https://www.analyticsvidhya.com/blog/2020/11/handling-imbalanced-data-machine-learning-computer-vision-and-nlp/ (https://www.analyticsvidhya.com/blog/2020/11/handling-imbalanced-data-machine-learning-computer-vision-and-nlp/)