# Quality Control Check for Proteomics Data

By Amril Prasad  Email: apr107@uclive.ac.nz

Supervisors: Charles Hefer & Evelyne Maes, AgResearch New Zealand

Nicholas Ward, Applied Data Science Department, University of Canterbury

## Table of Contents

# 1. Introduction

**1.1 Organization:** AgResearch is one the Crown Research Institutes which harness the power of Science to explore New Zealand's pastoral, agri-food and agri-technology value chains. AgResearch undertakes many local and international projects for benefiting the Agriculture sector and New Zealand.

**1.2 Project Description:**

The Proteomics team of AgResearch, based in Lincoln, New Zealand aims to build classification models for data generated by different Mass Spectrometers ( Impact HD and Impact_ii), which are generation apart. We also need to rank features according to their importance/contribution to the model. To understand data, first, we need to understand what Proteomics is. In brief, Proteomics is a study to detect protein expression patterns at a given time in response to a specific stimulus, but also to determine functional protein networks that exist at the level of tissue, cell or whole organism / compound[1]. The data given is from two mass spectrometers which are a generation apart. The Quality of resulting peptides is categorized as 'Good', 'Acceptable', 'Mediocre', 'Poor' and 'Pitiful'.

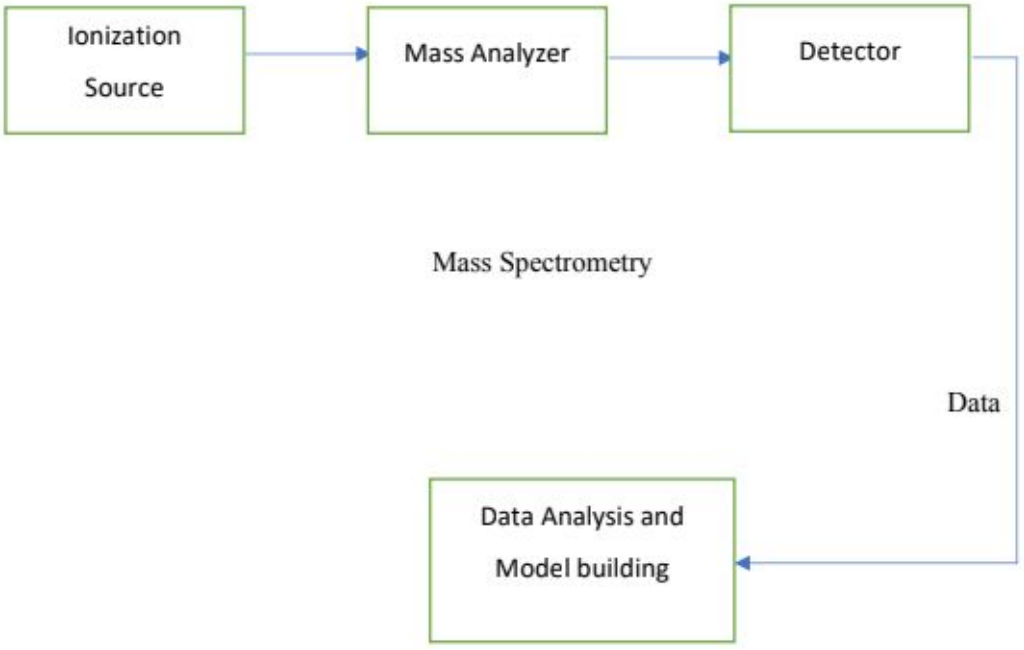


Figure 1 Flowchart of the experiment, data generation and analysis

### 1.3 Goals:

We need to test if there are certain Peptides, Modification or number of Peptides in a run (Peptides extracted in one go of the experiment ) which are unique in different Quality of Peptides.

Besides testing the above hypothesis, there are certain other goals which need to be explored in the project, which is listed below: -

Finding the features that help in detecting 'Quality' (target variable) from a large group of explanatory variables.

Rank the features which help in predicting 'Quality'.

Build classification model for data generated by Mass Spectrometers.

Assign '15 Minute Gradient' feature to Yes and build a classification model for impact II ( Mass Spectrometer) dataset.

Assign '15 Minute Gradient' feature to Yes and build classification model with features that Proteomics team use for classifying Quality of Peptides for Impact_HD ( data generated from Impact HD Mass Spectrometer) and Impact_ii ( data generated from Impact II Mass Spectrometer).

### 1.4 Constraints:

The main constraint in the project was, we must go somewhat blind towards the dataset as not much of the input was provided about what variable corresponds to what. This was part of the gameplay to get insights from the data from all possible angles. Another constraint was, the target variable (Quality of Peptides) was hand-curated by experts at AgResearch and it was difficult to rely on when there are 1,80,000+ observations.

### 1.5 Data:

The data was generated by Mass Spectrometers as a result of an experiment on Bovine Serum Albumin (BSA). The data generated by the experiment was in XML format. An XML parser was used to process data in '.csv' format. Some of the variables were hand-curated. The target variable 'Quality' was also hand curated by an expert intervention. Two datasets were

provided which were generated by two Mass Spectrometers (HD and impact II). Below is the tabular representation of both the data shapes: -

*Table 1 Data Information*

| Machine Name | Number of Variables | Number of observations |
|---|---|---|
| HD | 179202 | 53 |
| Impact II | 71240 | 53 |

The data lists out information about various experimental calculations involving mass, charge, particles per million etc. The data also lists out various data dumps, ion and molecular mass, Peptides, Sequence Coverage etc. The essential variables will be discussed later. The data consisted of a mix of Nominal, Categorical and Ordinal variables. The target variable which was to be classified as Ordinal in the order 'Good', 'Acceptable', 'Mediocre', 'Poor' and 'Pitiful'.

## 2. Methodology:

### 2.1 Impact_HD Data

As we have been given two datasets to analyse, we will first go with HD dataset. In the following section, we will discuss strategies undertaken to pre-process the data for modelling, feature extraction, feature selection, statistical analysis, visualizations and machine learning algorithms considered.

#### 2.1.1 Data Retrieval and Understanding

The data was retrieved in '.csv' format from the Proteomics team at AgResearch. The data is also available in GitHub repository which can be retrieved using a pull request over the server.

The dataset was loaded in R and was read in the form of a data frame. There were 179202 observations in the dataset. The dataset consisted of many dump folders location, directories and path. This was later confirmed that they are of no use and no additional information can be sought from there. Apart from this, the discussion went on some variables that need to be eliminated as they have a direct correlation with the target variable. Moreover, there is another separate process to hand curate the feature. We want a set of variables the spectrometer produces which can help in predicting Quality.

Below is a list of some important variables and what they mean: -

- 'Rt..min.' -The retention time of the peptide (described in minutes). This is the minute the peptide is eluting from LC (liquid chromatography) column
- IntCov..... – Intensity coverage as percentage
- MH..meas. - Measured mass of the ion
- Int. – The amount of Peptides detected by Mass Spectrometer
- Score - Score of the peptide given by the database search. The higher the score, the more confident the database search is that the MS/MS spectrum reflects a peptide has the actual sequence
- m.z.calc. - mass- over- charge ( calculated )
- Cmpd. - Compound - an automated number given during data analysis
- Mr.meas. - Measured molecular weight
- Peptide – The peptide found during that time
- Modifications – These are PTM ( Post translational modification), this gives a chemical compound when peptides react with certain chemicals and it happens at random.
- Run – It corresponds to a numerical number ( like a sequence) in accordance to experiment.

### 2.1.2 Data Cleaning

Some of the explanatory variables were removed at the request of Agresearch. With data variables left, some variables had near-zero variance like Boolean inputs or just 2 to 5 unique values. Figure 2 in appendix depicts the code and output of some of the cases where near-zero variance was observed. These columns were analysed then confirmed with supervisor and were dropped.

There were few columns which represented the Path file and storage location over the server, those were found to be of no use. Hence, they were dropped.

Some features had scientific notation ' Δ ' (delta). These columns were renamed by replacing the delta notation by 'delta'.

Two columns namely 'Score' and 'Scores' conveyed the same meaning. The difference was that Scores conveyed a little extra meaning which was not necessary for modelling. Figure 3

in the appendix depicts both the columns. Hence ‘Scores’ parameter was dropped. Along with it, columns which represented row numbers were also dropped.

***Discussion***

Discussion with the Proteomics team gave some insights about data like they manually render ‘Quality’ factor based on ‘RT min’; ‘Int’ and ‘SC’. We were also told ‘SC’ has a direct correlation with Quality and is processed in a different way and should not be considered. There were many meetings over the period of time and gradually new information was provided like ‘Cmpd.’ Feature need not be considered as it is a number given out by the machine, extracting a new feature from ‘Run’ and ‘peptide’ feature to form number of peptides per Run and unique Peptides in each ‘Run’, sorting class imbalance problem by combining few classes .

### 2.1.3 Data Strategies for impact_HD data

***Missing Data***

After getting descriptive statistics of data, it was observed that five features had exactly five missing values which account for less than 0.1%, the value was so small that on vis_miss plot it was not visible as shown in Figure 4.


Figure 4 : Viss miss plot depicting missing values that accounts for less than 0.1%

When NA values were dropped, there were five observations less. We can deduce that the missing values were present at the same position. The feature 'modification' had many values which were represented as "", which can be treated as missing values. The missing values seemed to be random and accounted for 39.57% (depicted in Figure 5).

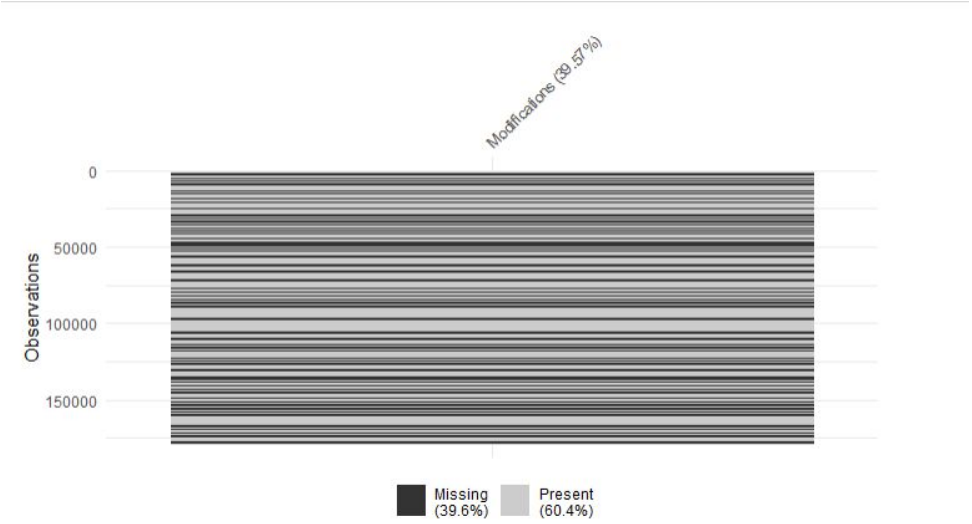


Figure 5: Missing values of Modifications feature.

The missingness was at random as Peptides reacted with certain chemicals at random to create these modifications. We decided to target encode Modifications along with missing values to see if these provide any insights.

***Outlier Analysis***

With help of descriptive statistics, it was evident that some features had many extreme values which were away from where 75% of the data present.

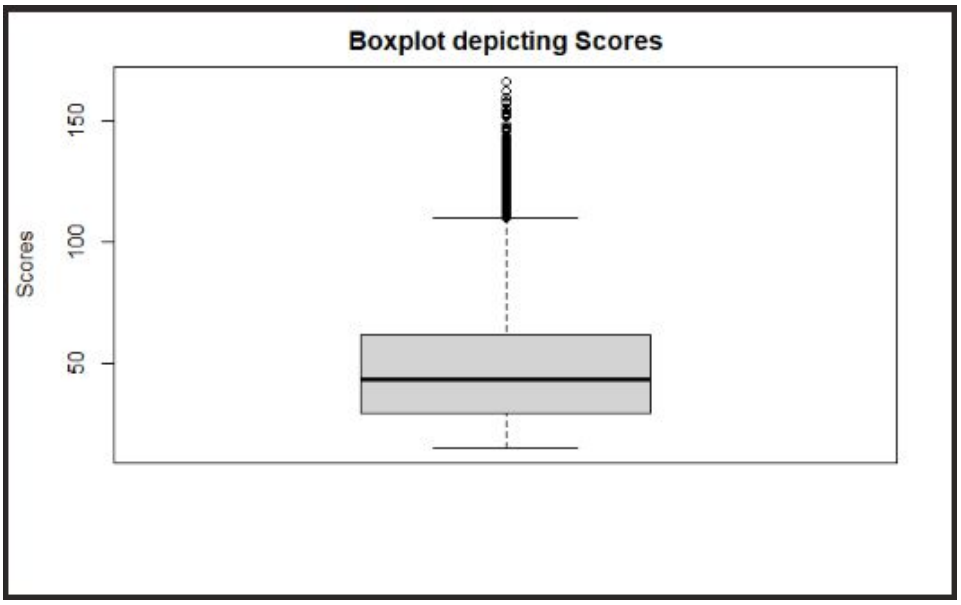


Figure 6: Boxplot of Scores feature showing some data points lying away from where the majority of data points are.

After discussion with the Proteomics team at AgResearch, it was advised not to remove those data points as they were experimental values. There are high chances that there must be a calibration issue with the machine which may explain some of the data points but not account for all. This implies that we can go for robust machine learning algorithms.

***Exploratory Data Analysis***

As per the hypothesis, we need to test the presence of Peptides and Modifications in Quality features. Figure 6 depicts Modifications presence in different Quality class and Figure 7 depicts Peptides present in each Quality class.

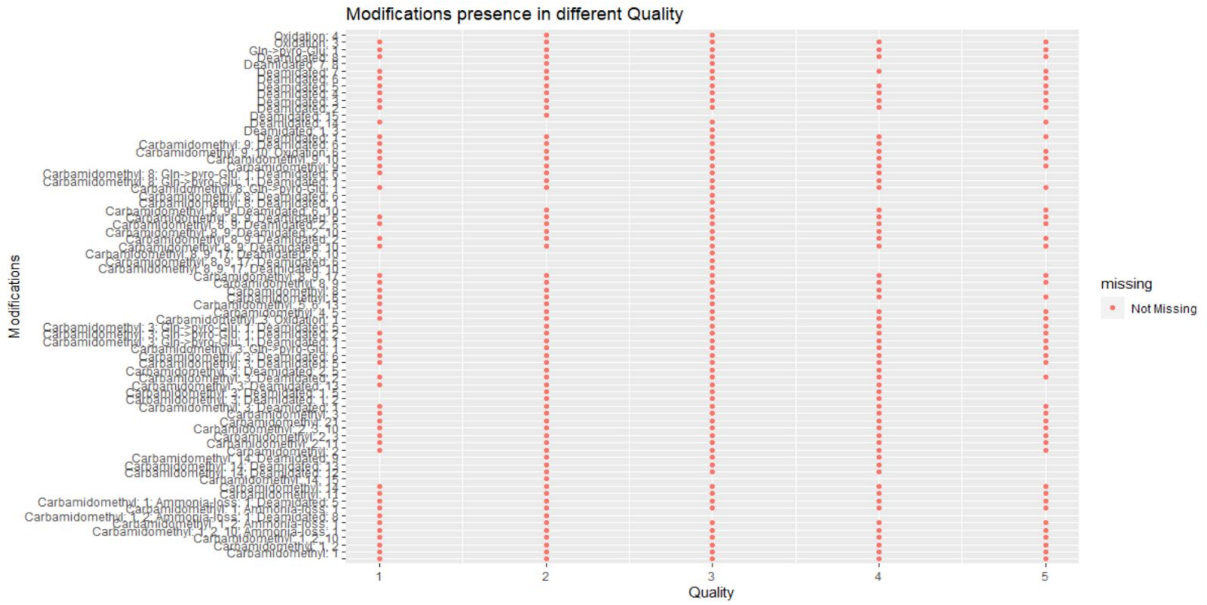


Figure 6 : Modifications presence in different Quality classes.

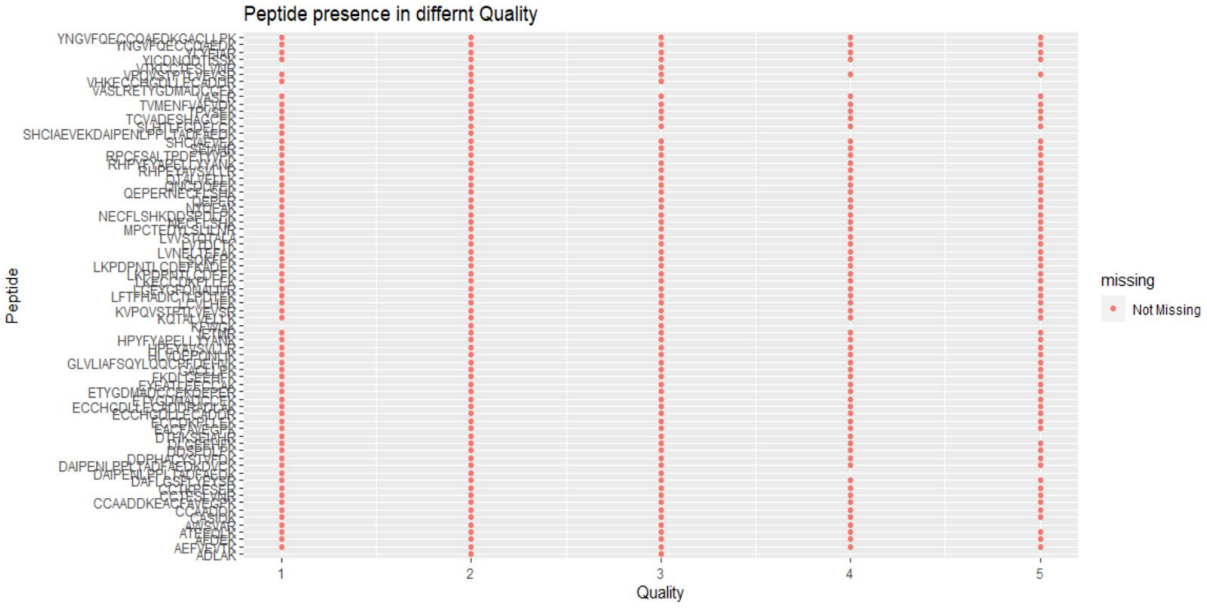


Figure 7 : Peptides presence in different Quality classes

Both the figures depict, more or less each Modification or Peptide were there in many Quality classes. It ruled out that Modifications or Peptides are unique to the Quality classes. Though from these graphs, we cannot find in what concentration they were present. Target encoder can be used which averages the target value by category.

The target variable ‘Quality’ was encoded in Integer type for analysing it with other variables. Though it is encoded in Integer, the values are still considered in Ordinal (1 being the best ). Since we need to build a classification model, the priority was to check if there is

an issue of class imbalance. Figure 8 depicts the number of observations present in each class of Quality which implied class imbalance problem. ‘Good’ and ‘Pitiful’ were negligible as compared to ‘Mediocre’. It was decided to merge ‘Good’ and ‘Acceptable’ together and ‘Poor’ and ‘Pitiful’ together.

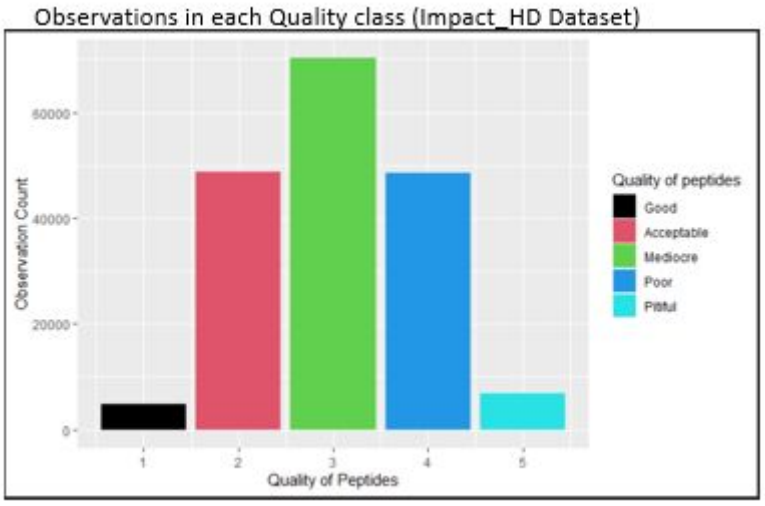


Figure 8 : Number of observations in each Quality class

Before merging, data was further explored using the five classes. Pearson’s Correlation visuals depicted few variables had poor to decent negative correlation and one variable had a decent positive correlation which cannot be used in modelling. Moreover, the correlation chart also depicted that the predictor data was highly correlated within itself. So, we need to use techniques like Principal Component Analysis (PCA) to counter this.

Various plots were plotted against Quality to see the trend, Score was one of the features which represented the confidence of the machine that MS Spectrum is representing correct Peptide Sequence. Figure 9 depicts the Concentration of ‘Score’ observations vs Quality.

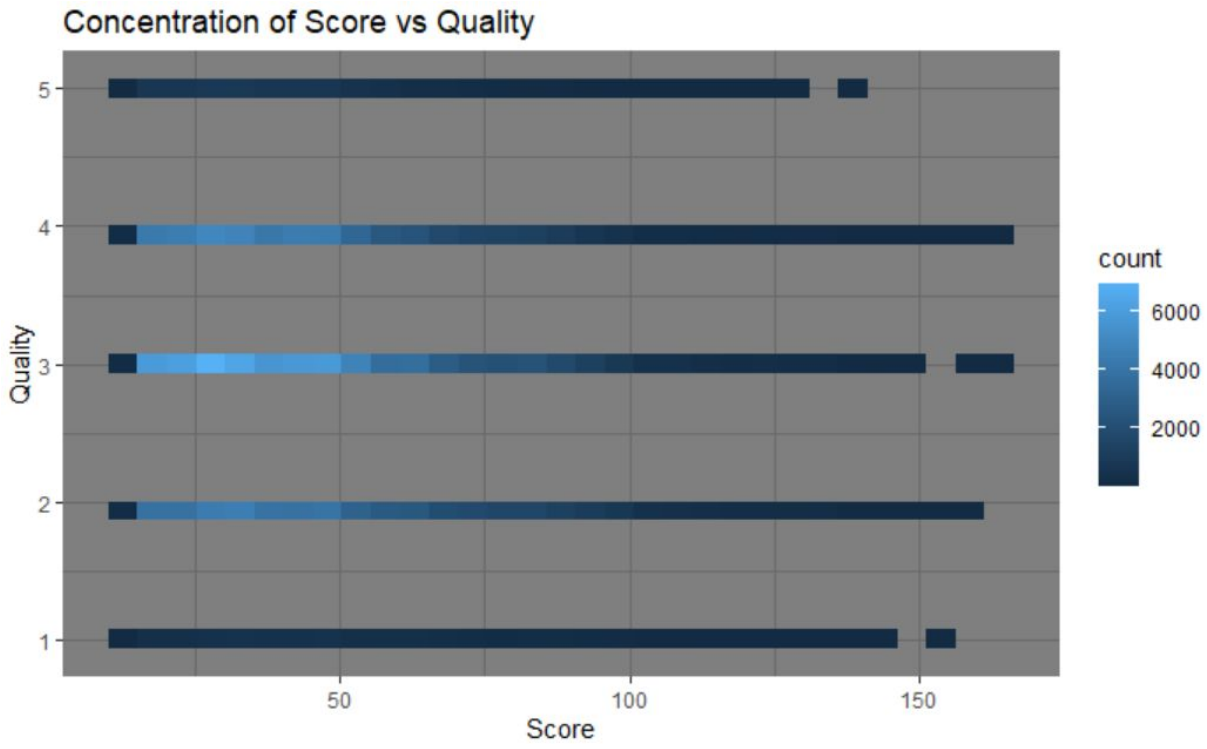


Figure 9 : Concentration Score Variables with regards to Quality. The lighter the colour, more the number of observations present

Random Forest Variable importance graph was also plotted to see which variables are important in predicting Quality. After confirming with the Proteomics team leader, it was advised not to take in account Molecular-level calculation and take just ion-based calculation.

***Feature Extraction: -***

When a sample goes through experiment in Mass Spectrometry, it is evident that some Peptides show up (Prior experience of the scientists). As the experiment progresses, issues with calibration happen due to internal and external factors which leads to non-detection or wrong detection of Peptides. Using 'Run' and 'Peptide' features, we were able to count the number of unique Peptides in each 'Run'. It had direct correlation with the Quality as more the Peptides in the Run, better was the Quality. Figure 10 depicts the stated relation. Upon discussion with the Proteomics team at AgResearch, it was found relevant.

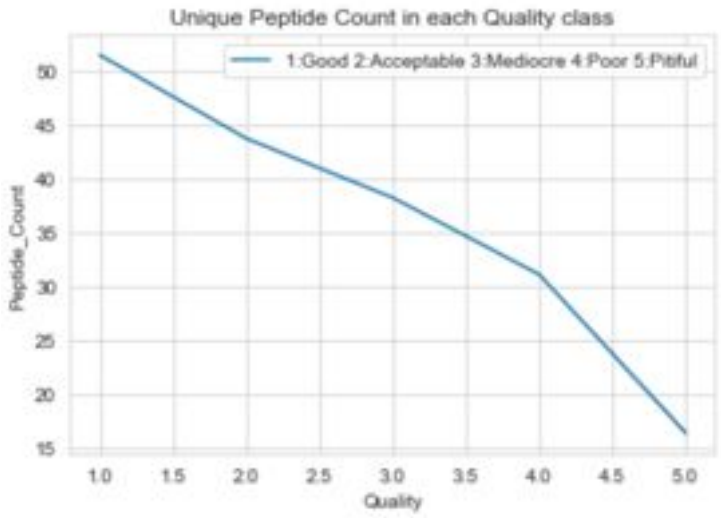


Figure 10 : Trend of unique Peptide count in a Run with Quality of Peptides

***Encoding Categorical variables: -***

To investigate furthermore into Peptides and Modifications, it was decided to encode them and validate its statistical significance. Since both the features suffered from high cardinality (>70), it was decided to go with target encoding and one hot encoding. Chi-squared test was implemented but both features had low scores. Even, they did not pop up in select K best features (10 features were selected). Figure 11 and 12 in the appendix represent code and chi-squared values.

***Class Balancing***

There are many ways to resolve the problem of class imbalance like Under sampling, oversampling which controls the number of observations present in train and test data to give an appropriate result[2] and we can also opt for regression analysis, where class imbalance problem can be less real. We decided to first go with a regression analysis on a five class problem. Lasso regression and Polynomial regression with different degrees was decided to be implemented. At the request of AgResearch, we were asked to merge the Quality classes as stated above to resolve the problem of class imbalance. The newly formed data had a good representation of each class. Figure 13 depicts the number of observations in three classes.

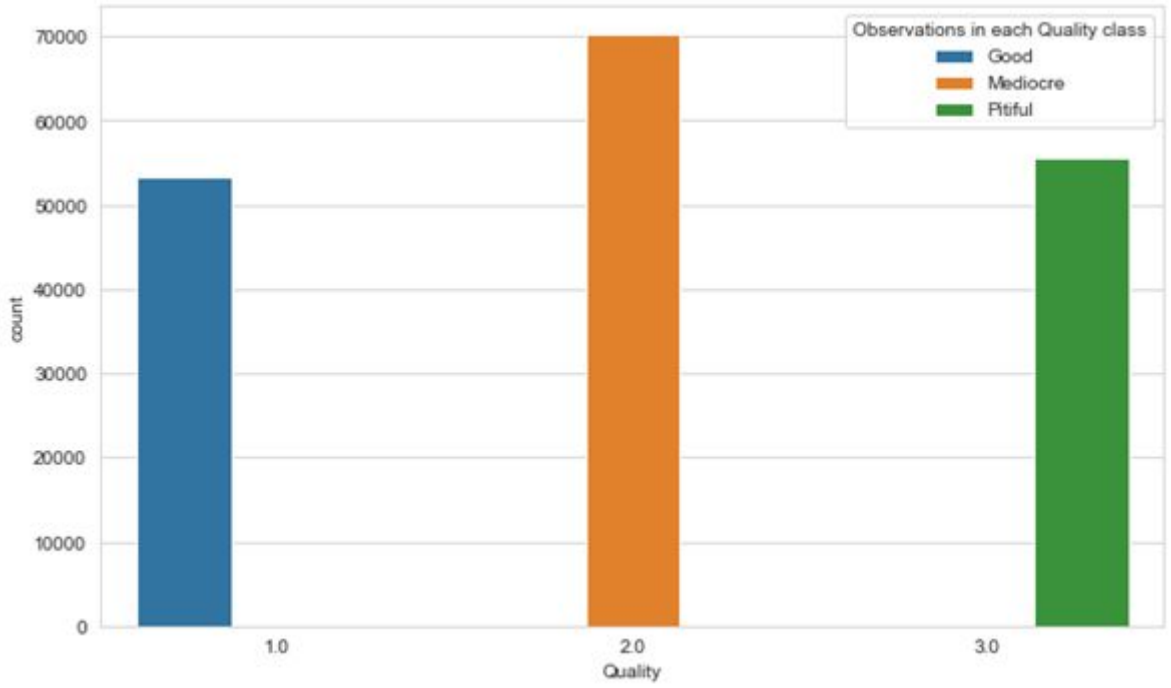


Figure 13 : Representation of newly formed three class Quality

## 2.2 Methodology for Impact_ii data :

The explanatory variables for impact_ii_data were the same as that of impact_hd data. We will discuss accessing, cleaning the data. Furthermore, different problems and issues will be addressed with solutions.

### 2.2.1 Data Retrieval and Understanding

The data was provided in a '.csv' format by the Proteomics team. The data is also available in github repository.

The data was loaded in Jupytr Notebook and '.csv' file was read in pandas Dataframe object. The first thing was to filter data on 15 Minute Gradient feature =' yes' , at the request of AgResearch. This data is to be used for analysis and modelling purposes. After filtering this, we were left with 70,887 observations. The data had repository dumps and various features for paths for data dumps which were not to be considered. As discussed with the supervisor at AgResearch, some features need not be considered as some are hand-curated and many are not relevant as they are molecular level calculations, random number generated by machine during the experiment.

Below is a list of some important variables and what they mean: -

- ‘Rt..min.’ -The retention time of the peptide (described in minutes). This is the minute the peptide is eluting from LC ( Liquid Chromatography) column
- ‘Length’ – Length of sequence of Peptide
- ‘ m’ – Measured mass of the ion
- IntCov. [%] - Intensity coverage as a percentage
- MH..meas. - Measured mass of the ion
- Int. - The amount of the peptide detected by the mass spectrometer
- Score - Score of the peptide given by the database search. The higher the score, the more confident the database search is that the MS/MS spectrum reflects a peptide has the actual sequence
- m.z.calc. - mass- over- charge ( calculated )
- Peptide – The peptide found during that time
- Modifications – These are PTM ( Post translational modification), this gives a chemical compound when peptides react with certain chemicals and it happens at random.
- Run – It corresponds to a numerical number ( like a sequence).

### 2.2.2 Data Cleaning

As discussed in section 2.1.2, some of the feature variables were removed. These included the data dumps, highly correlated hand-curated variable (‘SC’) and some features on-demand of the Proteomics team at AgResearch. Descriptive statistics depict some features that have near-zero variance. The number of unique elements was seen and irrelevant features were dropped.

Some features had mathematical notation ‘ Δ ’ (delta). These columns were renamed by replacing the delta notation by ‘delta’.

As discussed in section 2.1.2, Scores feature was also dropped.

### 2.2.3 Data Strategies for impact_ii data

***Missing Data***

Figure 14 depicts the missing data. Many observations appeared missing in random for Modifications feature, which can be understood because Post Translational Modifications of protein happen randomly. 353 observations were missing throughout. Moreover, the target

variable was also missing. These observations were dropped.

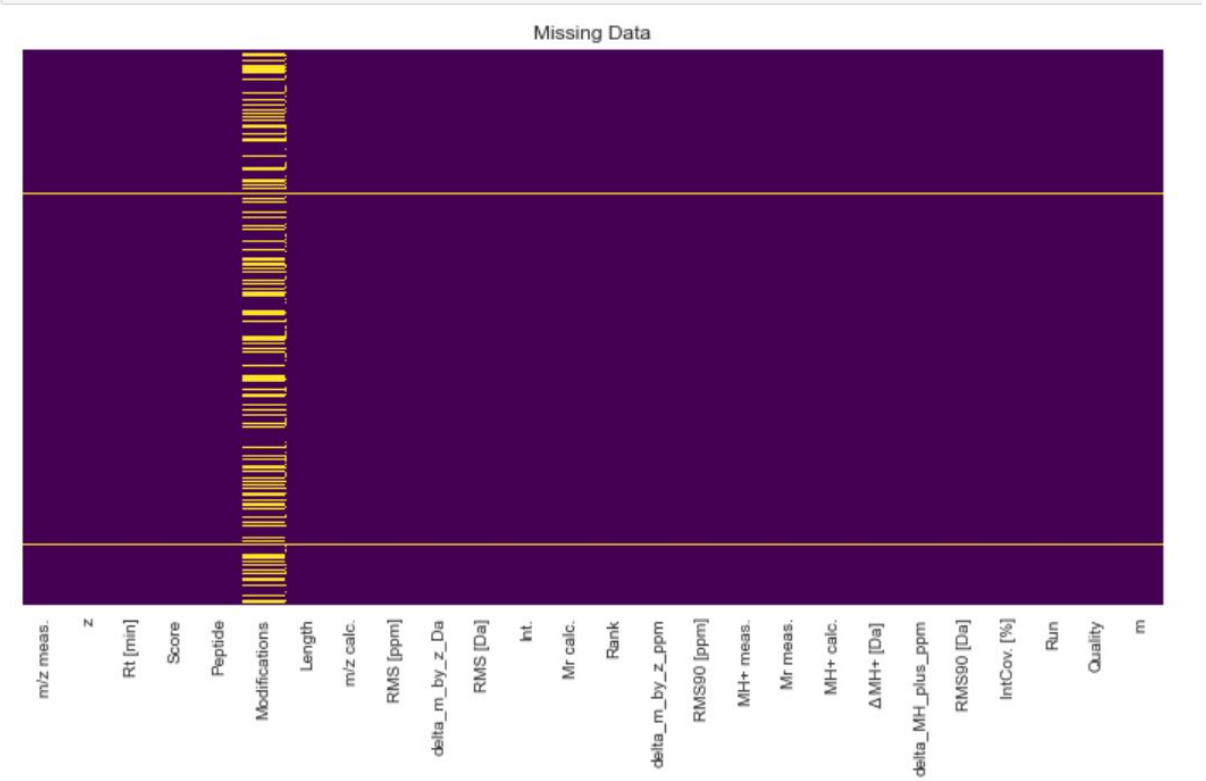


Figure 14 : Missing values in Impact_ii dataset

***Outlier Analysis***

The data correspond to values given out by machine during experiments. Though, for many features maximum value lies far from where 75% of the data value lies. With an earlier discussion regarding impact_HD dataset, it was not advised to remove data points, we shall not put any threshold for the data values.

***Exploratory Data Analysis***

As this is a classification problem, the problem of checking class imbalance was a priority. Figure 15 depicts the problem of class imbalance.

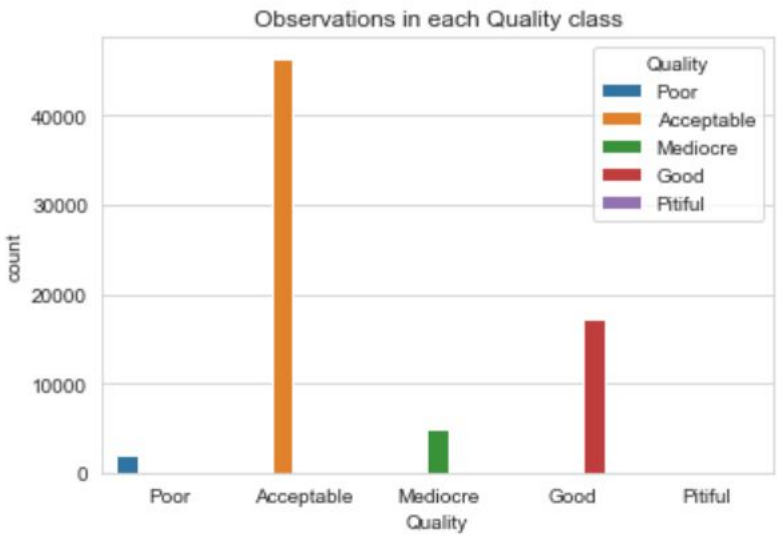


Figure 15 : Class imbalance problem ( Impact_ii dataset)

The regression analysis on five class problem did not gave satisfactory results on Impact_HD data and here the imbalance problem was even worse. So, it was decided not to go with regression analysis. The observations in the categories 'Good', 'Acceptable' were high as compared to 'Mediocre'. 'Poor' and 'Pitiful' had negligible observations as compared to 'Acceptable'. It was decided besides merging 'Poor' and 'Pitiful', we also need to go for techniques like under-sampling or oversampling. We decided to go with Random Oversampling as the difference between minority class and the majority was huge. Random Oversampling includes selecting random examples from the minority class with replacement and supplementing the training data with multiple copies of this instance, hence it is possible that a single instance may be selected multiple times[4]. We also need to investigate if Peptides and Modification are somewhat unique to Quality. After visualising, it was evident, they were present in almost every Quality class.

The correlation (Pearsons) chart showed that data was highly correlated within itself. To counter this problem, we will need Principal component analysis (PCA).

***Feature Extraction***

In the data, mass/charge ratio was provided. During the meeting, it was discussed that measured mass can be considered as a feature. The measured mass feature was extracted by using 'mass/charge' ratio and 'charge' features. There was not much difference between the mean and median of the newly formed feature 'm'. This feature also did not have correlation with the target variable. Figure 16 shows the relationship between measured mass and

Quality. The count of unique peptides per Run was also calculated the same way as in the Impact_HD data analysis.


Figure 16 : Measured mass of ion in each Quality class.

### *Encoding Categorical variables*

To investigate furthermore into Peptides and Modifications, it was decided to encode them and validate its statistical significance. Since both the features suffered from high cardinality (>70), it was decided to go with target encoding and one hot encoding. Chi-squared test was implemented but both features had low scores. Even, they did not pop up in select K best features (10 features were selected).

### *Class Balancing*

At the request of AgResearch, Poor and Pitiful categories were merged. After merging, we have applied Random OverSampling technique to increase the minority class observations by replicating them. The sampling strategy used was ‘minority class’. Considering 17155 and 46278 observations in ‘Good’ and ‘Acceptable’ class respectively, the number of observations in ‘Mediocre’ and ‘Poor’ were set at 15000. The aim was minority classes should represent at least 30% of the majority class. Figure 17 and Figure 18 represent code snippets and output depicting observations before and after Random Oversampling was performed.

# 3. Results

## 3.1 Impact_HD

At first, a random forest model was built with five classes after data cleaning. The class error in the training set was high (up to 77% class prediction error). The confusion matrix below depicts the results with the class error.

| | 1 | 2 | 3 | 4 | 5 | Class.error |
|---|---|---|---|---|---|---|
| 1 | 811 | 1606 | 731 | 163 | 10 | 0.7557964 |
| 2 | 57 | 16715 | 14924 | 2360 | 35 | 0.5096946 |
| 3 | 18 | 6491 | 35532 | 7328 | 38 | 0.2808307 |
| 4 | 5 | 2032 | 16389 | 15191 | 312 | 0.5522709 |
| 5 | 4 | 348 | 935 | 2478 | 1116 | 0.7713583 |

To overcome the class imbalance problem, this problem was also tested with regression analysis. The below table lists the results attained in different regression analysis.

| Machine learning Model | Metrics | |
|---|---|---|
| | R- Squared | RMSE |
| Lasso Regression | 48.20 | 0.88 |
| Polynomial Regression (degree = 2) | 24.31 | 0.87 |
| Polynomial Regression (degree = 3) | 24.10 | 0.87 |

These results were discussed with the supervisor and we concluded that even if they are not correctly classified in their respective classes, many of the observations are present in adjacent classes. Combining the classes will not just resolve class imbalance but we may achieve good results.

***Scaling***

We had used MinMax Scalar to bring data under one scale. MinMax Scalar was used because chi-squared estimation cannot handle negative values.

***Principal component Analysis***

Since the data was highly correlated, PCA was used to counter the problem of multicollinearity. Multicollinearity can cause use of explanatory variables unstable by the model. When these problems arise, there are various remedial measures we can take. Principal component analysis is one of these measures and uses the manipulation and analyzation of data matrices to reduce covariate dimensions while maximizing the amount of variation[3]. It also helps in dimension reduction as well. Model can be built taking a different number of components as a hyperparameter.

Then a Random Forest model was built using the newly extracted feature for unique peptide count and other 26 explanatory variables. The results were good both in the training set and test set gave good results with Precision and Recall as 80% and 79% respectively.

### 3.1.1 Feature Selection

After looking at Random Forest's important variable chart, selectKbest approach and discussion with Proteomics team, we concluded to first go with the following set of features.

- Rt..min.
- IntCov. [%]
- MH..meas.
- Int.
- Peptide_Count
- Score
- m.z.calc.
- Length

We had to remove some of the important features suggested by random forest variable importance graph and chi squared test like ‘Cmpd.’. ‘Cmpd.’ is  a random number generated by the machine and all the features that accounted for Molecular level calculation were also removed.

### 3.1.2 Algorithms Used

- Random Forest
- Support Vector Classifier
- MLP Classifier

### 3.1.3 Modelling results

| Algorithm Name | Constraints | Precision (in %) | Recall (in %) |
|---|---|---|---|
| SVC | Kernel =rbf<br>PCA(n_components = 3) | 79 | 75 |
| SVC | Kernel =rbf<br>PCA(n_components = 5) | 79 | 75 |
| SVC | Kernel = poly<br>Degree = 3 | 78 | 75 |
| MLP Classifier | hidden_layer_sizes=(256,128,64,32)<br>Activation = 'relu' | 77 | 76 |
| Random Forest | 'n_estimators' : 600<br>'min_samples_leaf' : 3<br>'min_samples_split' : 2<br>'max_features' : [None] | 77 | 75 |

GridSearchCV was implemented both in SVC and random Forest Classifier to get best estimates.

In SVC, we iterated through the following conditions: -

Kernel = 'poly'

Degree= [3,4,5]

The best parameter was with degree = 3

In Random Forest classifier, following set of hyperparameters were given: -

'n_estimators' : [400,600],

'min_samples_leaf' : [3,4],

'min_samples_split' : [2,4,5],

'max_features' : [None]

The best parameters were as following: -

'n_estimators' : 600

'min_samples_leaf' : 3

'min_samples_split' : 2

'max_features' : [None]

Using the above best parameter for Random Forest, the model was overfitting. The discrepancy between train data result and Test data result was high.

In MLP classifier, we iterated through the following conditions:-

hidden_layer_sizes=(256,128,64,32)

Activation = ‘relu’, ‘tanh’

‘relu’ activation function performed better than tanh.

***Best Performing Model***

**Support Vector Classifier :-** Support Vector Machines is a supervised machine learning algorithm. It can be used both for regression as well as classification. It plots each data item as a point in n-dimensional space (where n is number of features in the dataset) with the value of each feature being the value of a particular coordinate. Moreover, with the rbf kernel, the model can also learn the non linear relationship of the data. So, the best result was with the SVC using the kernel as ‘rbf ‘ and the number of PCA components as 3 or 5.

The team at AgResearch were convinced with these results.

### 3.2 Impact_ii

***Scaling***

We had used MinMax Scalar to bring data under one scale. MinMax Scalar was used because chi-squared estimation cannot handle negative values.

***Principal component Analysis***

Some explanatory variables were highly correlated to each other, which causes the problem of multicollinearity while modelling. Principal Component Analysis not only solves the problem of multicollinearity but also does dimensional reduction.

### 3.1.2 Feature Selection

After looking at Random Forest's important variable chart, selectKbest approach and discussion with Proteomics team, we concluded to go with the following set of features.

- Rt..min.
- RMS90 [Da]
- Peptide_Count
- Score
- m
- IntCov. [%]

### 3.2.2 Algorithm Used

Random Forest

### 3.2.3 Modelling Results

The data were modelled using Random Forest Classifier. We tried giving n_estimators (hyperparameter) as 100, 200 and 300. In all the cases, both Precision and Recall were 90%.

In Random Forest classifier, the following set of hyperparameters were given: -

'n_estimators' : [400,600],

'min_samples_leaf' : [2,3,4],

'min_samples_split' : [2,4,5],

'max_features' : [None]

The best parameters were as following: -

'n_estimators' : 600

'min_samples_leaf' : 4

'min_samples_split' : 2

'max_features' : [None]

But using the best parameters, given by GridSearchCV for the Random Forest model was overfitting. So, going for the simplest model, that is Random Forest with no hyperparameter

tuning was considered to be the **best** and Agresearch's Proteomics team was also satisfied with the result.

### 3.3 Modelling result with parameters suggested by Agresarch

At the request of the Proteomics team leader, following variables were chosen for modelling Impact_HD and Impact_ii dataset.

- Rt..min.
- SC
- Peptide_Count
- Int

Here, we had two features 'Peptide_Count' and 'SC' which had strong correlation ( > 0.83) with target variable 'Quality'.

Random Forest classifier was used for modelling after countering class imbalance for both the data as done previously. Both Precision and Recall were 100% for both training and test data.

## 4. Future Work

There is a possibility of improvement in XML parser. Building a more accurate XML parser may result in getting additional features in '.csv' format. A classifier model can be built using data of both the Mass Spectrometer.

# References


Berghmans, Eline. *Mass Spectrometery imaging combined with top-down proteomics to predict a more accurate immunotherapy response in non-small cell lung cancer patients*. PhD Thesis. Belgium , 2020.

Upasana. *https://www.analyticsvidhya.com/blog/2017/03/imbalanced-data-classification/*. n.d. Webpage.

Perez, Lexi V. "Principal Component Analysis to Address Multicollinearity." 2017.

Pykes, Kurtis. *https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf*. 2020. Web Page.

# APPENDIX

```r
```{r}
unique(data_hd$OK)
unique(data_hd$z)
unique(data_hd$X.Cmpds.)
unique(data_hd$Injection.Amount)
unique(data_hd$AltPep.)
unique(data_hd$Rank)
unique(data_hd$P)
```

 [1] FALSE  TRUE
 [1] 2 3 1 4 5
 [1] 0 2 1 4 3
 [1]  50 100
 [1] 0 1 2
 [1] 1 2 3
 [1] 0 1
```

Figure 2 : Near zero variance depicted in some explanatory variables

| Scores <chr> | Score <dbl> |
|---|---|
| 31.0 (M.score:31.0) | 31.0 |
| 50.2 (M.score:50.2) | 50.2 |
| 20.6 (M.score:20.6) | 20.6 |
| 28.2 (M.score:28.2) | 28.2 |
| 53.1 (M.score:53.1) | 53.1 |
| 15.1 (M.score:15.1) | 15.1 |

Figure 3 'Score' and 'Scores' features

```python
sf = SelectKBest(chi2, k='all')
sf_fit = sf.fit(temp_X, temp_y)
# print feature scores
for i in range(len(sf_fit.scores_)):
    print(' %s: %f' % (temp_X.columns[i], sf_fit.scores_[i]))
```

Figure 11 Code snippet of SelectKBest method with chi2

```
Cmpd.: 106.912658
m.z.meas.: 93.421370
z: 7.979427
Rt..min.: 497.458447
Score: 144.449535
X.Cmpds.: 20.974895
Start: 13.553970
End: 13.768615
Length: 126.697985
Mr.calc.: 117.194747
delta_m_by_z_ppm: 16.171471
RMS90..ppm.: 37.914238
MH..meas.: 117.194348
Mr.meas.: 117.194343
m.z.calc.: 93.421618
MH..calc.: 117.194733
delta_m_by_z_DA: 5.539784
delta_m_z_plus_DA: 5.387293
delta_m_z_plus_ppm: 16.172932
RMS90..Da.: 46.621628
RMS..ppm.: 55.553780
RMS..Da.: 63.625702
Int.: 72.083348
IntCov.....: 663.108025
encoded_peptide: 33.941953
encoded_mod: 11.257560
Peptide_Count: 212829.480575
```

Figure 12 Chi squared scores for variables

```
In [390]: y.value_counts()

Out[390]: 2.0    46278
          1.0    17155
          3.0     4869
          4.0     2230
          Name: Quality, dtype: int64
```

Figure 17 : Count of observations before Random OverSampling

```
In [393]: y_over.value_counts()

Out[393]: 2.0    46278
          1.0    17155
          3.0    15000
          4.0    15000
          Name: Quality, dtype: int64
```

Figure 18 : Count of observations after Random OverSampling