Part I

Challenge 3: Chemicals

2 Introduction

This part of the report is concerned with Challenge 3: Detecting Contaminated Chemicals, also known as Chemicals.

2.1 Structure of Part I

Chemicals begin with an Introduction, which includes Background, Dataset Summary, Aims, Challenges, and Main Findings, as well as Fundamental Notation. The next section is concerned with Data Engineering and EDA. In EDA section, 3 observations are given, which will provide a base for the modeling. Next, the Methods and Result section is provided. It includes 3 models used. Every model is a logical development of the previous one. At the end, a Discussion is given.

2.2 Background

Many substances can be labeled as chemicals. These include pharmaceuticals, fertilizers, water, pesticides, and many other chemical compounds. The purity of a chemical is crucial in many industries, such as medicine, agriculture, and water management.

In the medicine industry, pharmaceuticals are used to treat a disease or to ease the discomfort of the patient [1]. Hence, appropriate medicine at an appropriate dose must be delivered to a patient. When a medicine contains an unfortunate impurity, especially in high concentration, it might lead to a worsening of the patient's condition, creating additional health problems, or even death.

Consider the contamination of steroid injections in the USA in 2012. Steroid injections (called *Methylprednisolone Acetate*) were contaminated with a pathogen called *Exserohilum Eostratum* [2]. These steroid injections are used to treat blood disorders, immune system disorders as well as pain and swelling occurring with joint disorders (such as arthritis) [3]. The contamination lead to *Fungal Meningitis*; a life-threatening infection that causes swelling around the brain and spinal cord [2]. There were 751 patients affected, across 20 states of the USA. The contamination caused the death of 64 people [2]. Patients exposed to very high dosages were more likely to develop serious health issues.

The above example shows that the detection of contamination, type of contamination, and concentration is crucial in many industries. To achieve this, a device measuring wavelength spectrum is used. There are many different wavelengths, measured at so-called channels.

2.3 Dataset Summary

The dataset has been provided by an unknown real company and all details that might have led to the identification of such a company have been anonymized. Hence, the name of the company, the industry, and the actual contaminants are not known. It is assumed that the dataset contains true readings that were collected appropriately and so the dataset can be trusted. The dataset is split into two parts: *Train* and *Test* data. Train data is used to fit models (and estimate validation error), while Test data is used to check for prediction accuracy.

Train data has 8 columns and 92 rows, corresponding to 8 variables and 92 observations respectively. It can be visualized in Table 1.

Impurity.Percent	Impurity.Type	I	II	III	IV	V	Temp
2.69	A	103.9	14.4	9.5	5.0	49.9	31.42
...	...	...	...	...	...	...	...
7.20	N	97.7	14.8	10.1	6.5	64.6	32.49

Table 1: Visualization of the Train data

There are 2 dependent variables: Impurity.Percent and Impurity.Type, which represents the impurity percentage (by total weight) and the type of impurity respectively. The Impurity.Percent values in the Train data are continuous numbers, with accuracy to two decimal places. The Impurity.Percent values in the Train data are in the 1.86 to 7.94 range (there are no pure chemicals in the Train data, see section 2.4). The Impurity.Type is a categorical variable with 13 classes. Classes are denoted as letters A to N (with no I). The number of observations per class is summarized in Table 2.

Class	A	B	C	D	E	F	G	H	J	K	L	M	N
Number of cases	8	9	5	5	8	7	5	6	7	7	8	10	7

Table 2: Number of observations per class in Train data

Hence, there are 5 to 10 observations per class (see section 2.4). There are 6 independent features: readings from 5 channels (I-V) and temperature variable (Temp). Readings from channels are all continuous numbers with an accuracy of 1 decimal place and temperature variable has an accuracy of 2 decimal places.

2.4 Aims, Challenges and Main Findings

2.4.1 Aims

As described in section 2.3, Train data contains the impurity percentage and impurity type, while Test data does not. The aims are to:

1. Predict whether the chemical is pure
2. Predict the type of contamination (if not pure)
3. Predict the percentage of such impurity

Where predictions are based on readings from channels and temperature. Since impurity percentage is a quantitative variable, while impurity type is a qualitative variable, the analysis combines regression and classification. Therefore, the mathematical aim is to minimize the total error metric from equation (2.4.1).

$$\sum_{i=1}^{100}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{100} I(g_i \neq \hat{g}_i) \tag{2.4.1}$$

Where y_i and $\hat{y}_i$ denote the actual and predicted values of impurity percentage respectively. Similarly, g_i and $\hat{g}_i$ are the actual and predicted types of contamination (or pure chemicals). The first part relates to regression and denotes the sum of squared differences between the actual and predicted values of impurity percentage. This error can take values in the range $[0, \infty)$. The second term is the sum of all misclassification errors, where $I(g_i \neq \hat{g}_i)$ takes the value 1 if the observation is misclassified and 0 otherwise. The classification error takes values in the $[0, 100]$ range, where the error of 100 means that all observations have been misclassified. This will become important to identify problems with model performances in further sections.

2.4.2 Challenges

There are a few challenges that make this task interesting:

1. The problem combines regression and classification
2. There are no pure classes in Train data, but there are some in Test data
3. The number of observations per class is small (especially for certain classes)
4. The score combines regression and classification, but the proportion is not given
5. The skewness between Train and Test datasets is different

The first problem is relatively unique and requires a careful decision, whether to regress first and then use classification, do it the other way round, or use a mixture of both. The second one is a significant challenge that is difficult to solve and requires ingenuity. The third problem forces us to consider methods that work well with few observations per class. The fourth one will require to think carefully, about which part of this challenge (regression or classification) is a problem in terms of accuracy. The last one may lead to the problem of underfitting. All methods to overcome these challenges are described in appropriate sections.

2.4.3 Main Findings

The methods used include Logistic Regression, Lasso, Linear Discriminant Analysis, Manual Decision Boundaries, General Linear Models, as well as Generalised Additive Models. As a result of logical model development, the score has been improving every week (with one exception), indicating improvement in the methods used. The final score is one of the top of all groups, in all weeks. The analysis of the methods used confirmed that classification should be carried out first, models should be fitted based on classes, data is linear within classes and pure chemicals can be identified through regression. The main limitation of this approach is looking at marginal plots only.

2.5 Fundamental Notation

It is crucial to establish a basic notation that will remain constant for Part I of this report. Note that i and j are dummy indices corresponding to an observation number (either Train or Test) and parameter number respectively.

- y_i : true value of impurity percentage
- g_i : true type of impurity (or purity)
- $x_{i1}, x_{i2}, ..., x_{i6}$: readings from channels I, II, ..., V and temperature
- $\hat{y}_i$: predicted impurity percentage
- $\hat{g}_i$: predicted impurity (or purity)
- Q: arbitrary class (or set of classes)
- y_i^Q, ϵ_i^Q : true impurity percentage and random error for a model corresponding to Q
- β_j: parameter (β_0 is the intercept)

This notation allows to differentiate between actual/predicted values, dependent/independent variables and in-between models. Any additional notation will be defined at appropriate sections.

3 Data Engineering and EDA

This section describes Data Engineering and Exploratory Data Analysis (EDA) carried out for Chemicals. The EDA revealed interesting properties of the dataset. These are referred to as *observations*. Note that the structure of the data is described in section 2.3.

3.1 Data Engineering

To make the programming easier, it is useful to rename some variables. Therefore, Impurity.Percent will be now called *Percent* and Impurity.Type becomes *Type*. As a part of data cleaning and preparation, no missing values have been found in either, the Train

or Test data. Impurity type has been changed to a factor for regression purposes. Both, Train and Test datasets are structured in a clean way. Therefore, no other changes had to be made. During EDA we found that data is positively skewed for most of the features so square root transformation is done to transform a non-normal variable into a more normal distribution. This is done to meet the normality condition required for various statistical methods.

3.2 EDA: Observation 1

Many plots have been analyzed for the EDA. Those revealed that certain classes behave differently from others, as shown in Figure 1 below.

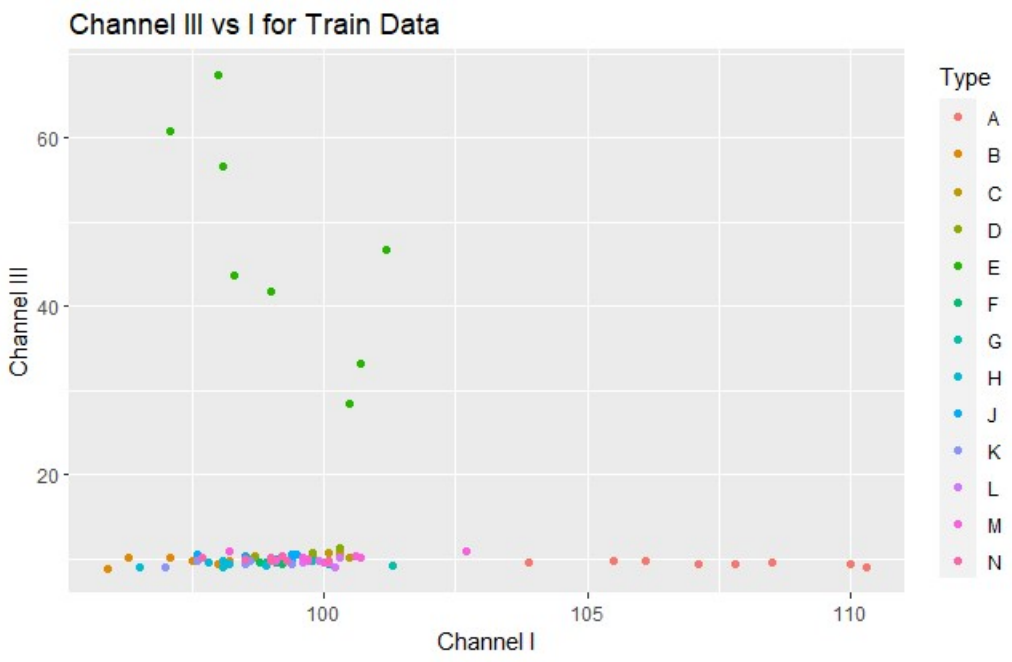

Figure 1: Scatterplot of Channel III vs I for Train Data

Clearly, class A (red) and class E (green) behave differently to others. This is also the case for other classes, for different combination of variables. This will become very useful in both, classification and regression.

3.3 EDA: Observation 2

Next observation is concerned with linearity in the data. Figure 2, represents percent impurity vs channel II for chosen classes.

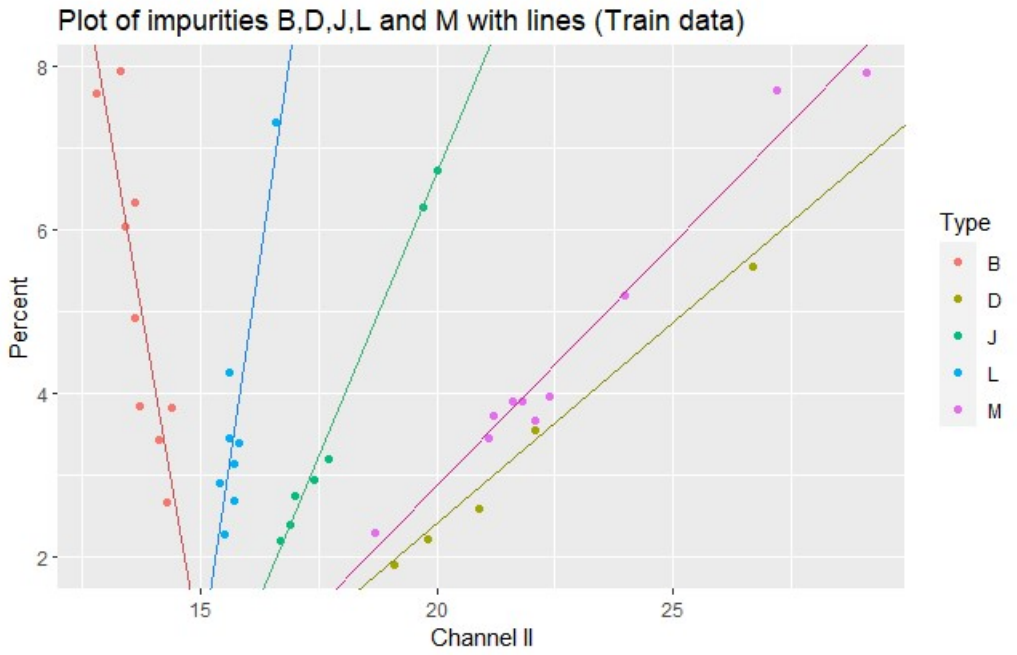

Figure 2: Plot of 5 classes with lines for each class

It is clear that (for these classes) there is a linear relationship between impurity percent and channel II. In addition, there are different slopes and intercepts for each class. This will become very useful in creating regression models. However, if the exact same classes are plotted with temperature as independent variable (as in Figure 3), the relationship is no longer linear.

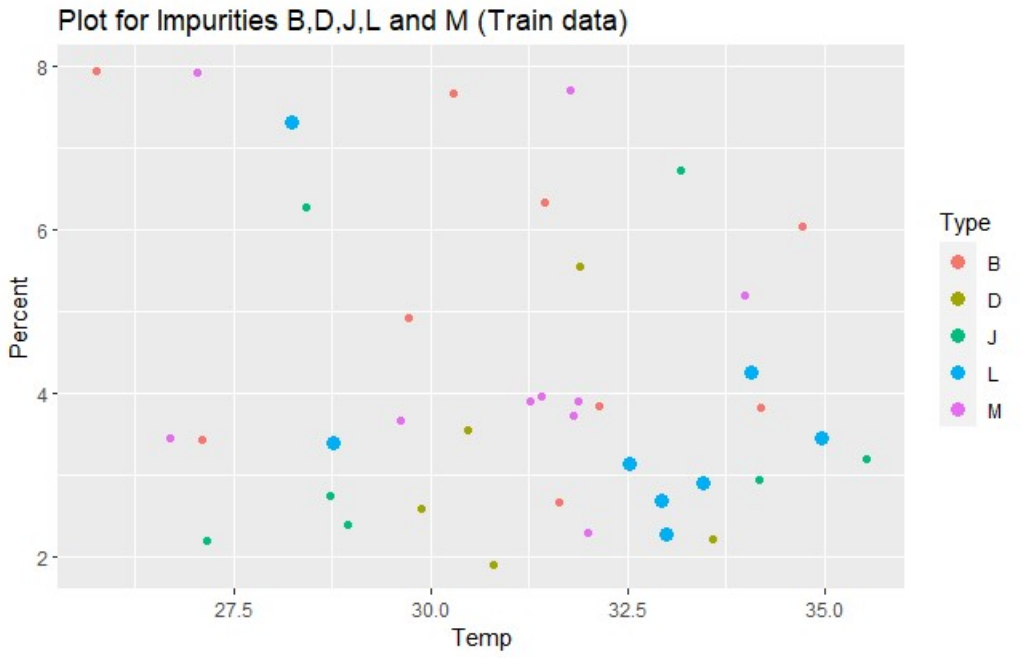

Figure 3: Plot of 5 classes with class L highlighted

It is clear, that there is no linear relationship between percent impurity and temperature (for class L). Not only the relationship is not linear; it is difficult to identify the type of non-linearity. Quadratic model is not appropriate. Neither is cubic, or exponential. This is the case not only for class L, but also for other classes and different features, as described in next paragraph.

The above two plots relied on visual assessment of linearity. One can go further and use Pearson's correlation coefficient r_{ij}, which measures the degree of linear correlation between variables i and j. For purposes of this challenge, a relationship is called linear if $|r_{ij}| \geq 0.7$. This provides objective rule of determining, what is meant by a linear relationship. For non-linear relations, it is difficult to identify the type of non-linearity. Due to the fact that it is inappropriate to use linear models for such cases and it is impossible to detect the type of non-linearity, non-linear cases will be considered as if there is no relationship. Relationships between variables for all classes, are summarised in Table 3.

Impurity	A	B	C	D	E	F	G	H	J	K	L	M	N
I	●	●	○	○	●	●	●	●	●	●	○	○	●
II	●	●	●	●	○	○	●	●	●	●	●	●	○
III	●	○	○	○	●	○	●	○	○	○	○	○	○
IV	○	●	○	●	●	●	●	●	●	●	●	●	●
V	●	○	●	●	●	○	○	●	●	●	●	●	●
Temp	●	○	●	○	○	●	●	●	○	●	○	○	○

Table 3: Type of relationship (within classes) between Independent variable and features

Here, ● denotes linear relation between impurity percent and independent variable and ○ means no clear relationship. This table shows that certain features are very linear (in relation to impurity percent and within classes) than others. For example, 85% of classes in Channel IV is considered linear, while only 23% for Channel III. In addition, certain classes are linear over more features than other classes. For example, class A is linear over 5 features, while class N is linear in only 3 features. These findings motivate using linear models for regression, with some features selection.

3.4 EDA: Observation 3

Due to the fact that there are no pure chemicals in Train data, but there are some in Test data, one needs a way of determining the pure chemicals. This problem is very difficult to solve and very often leads to inaccurate results. However, there are some approaches that can help. The first is to look as basic summary statistics. The basic summary statistics for each feature in train are summarized in Table 4.

Variable	I	II	III	IV	V	Temp
Minimum	95.90	12.80	8.80	4.90	47.00	23.62
Mean	99.81	16.41	13.075	10.20	54.90	31.12
Maximum	110.30	29.10	67.5	34.70	75.6	35.53

Table 4: Summary Statistics for Independent Variables in Train data

Clearly, there is a difference in readings between channels and temperature variables. The next logical step is to compare Train and Test data. The structure of Test data is similar to Train data, with some adjustments. Test data has 6 independent variables and 100 observations. There are no dependent variables, as these are variables to be estimated. However, it is known that Test data also has some pure classes (denoted as X). When a chemical is pure, the impurity percentage should be 0. The basic summary statistics for each variable are summarized in Table 5.

Variable	I	II	III	IV	V	Temp
Minimum	94.70	13.10	8.60	4.80	47.20	22.16
Mean	99.24	16.4	12.34	10.305	54.48	30.52
Maximum	105.30	29.6	68.50	35.20	75.6	36.47

Table 5: Summary Statistics for Independent Variables in Test data

There are some differences in summary statistics between Train and Test data. However, in general, they are similar. This motivates looking at a different metric.

One of them is based on overlaying Train and Test data on top of each other and looking for differences. The intuition tells that there must be a structural difference between Train and Test data. If there is a clear cluster of observations in Test which is different to Train, the cluster might indicate pure chemicals. Consider Figure 4 below.

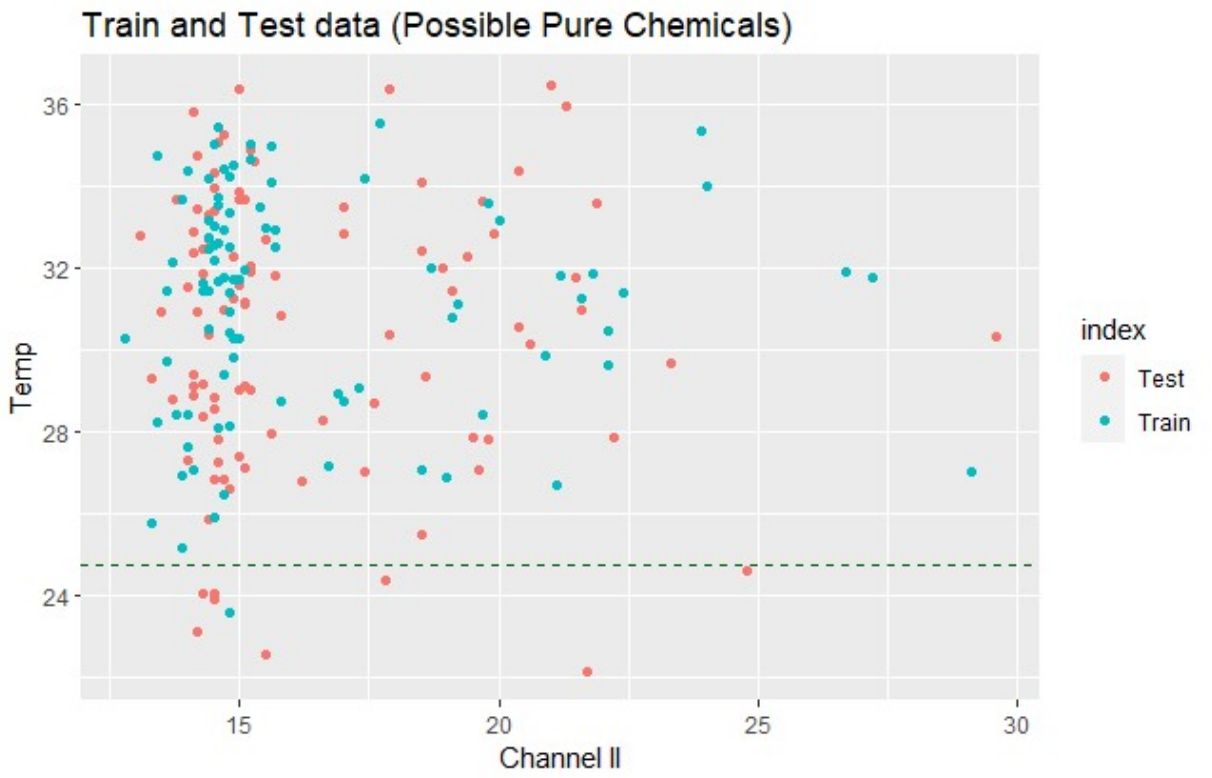

Figure 4: Plot of Temperature vs Channel II indicating possible pure chemicals

Note that *index* variable is to differentiate between Train/Test data. The Train and Test data are similar in structure. However, there is one cluster that might indicate pure classes. Notice that lower values of Temperature variable (for Test data) is slightly different that Train data. The green dotted line denotes Temperature of 24.5. Values below this line might indicate pure classes. This is due to the fact that this cluster does not have a corresponding observations in Train data. There is one outlier in Train data, which should be ignored.

4 Method and Results

The EDA is previous section motivated carrying out classification first and then using regression (based on classes). This section shows model progression in three stages, from basic to intermediary to the final stage, for both regression and classification separately. At each stage, the motivation behind the model is provided and shortcomings are addressed. The limitations will provide a motivation for the next stage. Although there were five different models for each week's prediction, only three are going to be discussed.

4.1 Classification

4.1.1 Initial Modelling

The Train data has been split further, into Train and Validation data, using random cluster sampling. Train is used to fit models, while Validation is used to check predictions. The base model for classification is the Logistic Regression. Even though it has 'regression' in its name, it can also be used for classification. This method is simple, yet powerful. In addition, it uses linearity findings from EDA. Logistic Regression [4] models the log-odd of an event as a linear combination of five independent wavelengths and one temperature feature, as in equation (4.1.1).

$$Pr(g_i = Q|X = x) = \frac{1}{1 + \sum_{l=1}^{Q-1} exp(\beta_0 + \sum_{j=1}^{6} \beta_j x_{ij})} \tag{4.1.1}$$

Where Q denotes a particular class. After training the model, the 10-fold cross-validation [4] is used to ascertain the goodness of fit. The accuracy of classification obtained was 0.83 ($\pm$ 0.103). The (relative) poor performance of the model can be attributed mainly to the fact that redundant features are also used to carry out classification. Redundant features include channel III and temperature, as found by Akaike Information Criteria (AIC). Including those features (and due to noise in the data) makes it harder for Logistic Regression to learn the true decision boundaries between classes. Other reason can be very

few data points per class. Classifying pure chemicals is based on regression described in further sections. This model failed to classify any chemicals as pure. This problem is being corrected in next stage, the Intermediate Modelling, where pure chemicals are identified, based on cluster of observations found in EDA.

4.1.2 Intermediate Modelling

The intermediate model aims to improve on the issues from the initial model, as well as apply observations from EDA. The overview of the modeling is as follows.

- Treat classification and regression problems separately.
- Join together similar classes. If a class is significantly different from others, do not join it
- Classification is carried out, by using Linear Discriminant Analysis (LDA) for the cluster of classes. For a single class, a manual decision boundary is created.
- Pure chemicals are classified based on manual decision boundary for the Temperature variable

Train data has been split into further Train data and Validation data. Exactly 67 observations were put into Train data and 25 to Validation data, using *sample* function. This function randomly selects indices. For reproducibility, the seed was set to 57. Due to the space constraints of the report, it would be infeasible to explain the procedure used for clustering of classes separately. Instead, examples are given to cover all ranges of methods, and a summary for all is given at the end. Consider Train data plotted in Figure 5.

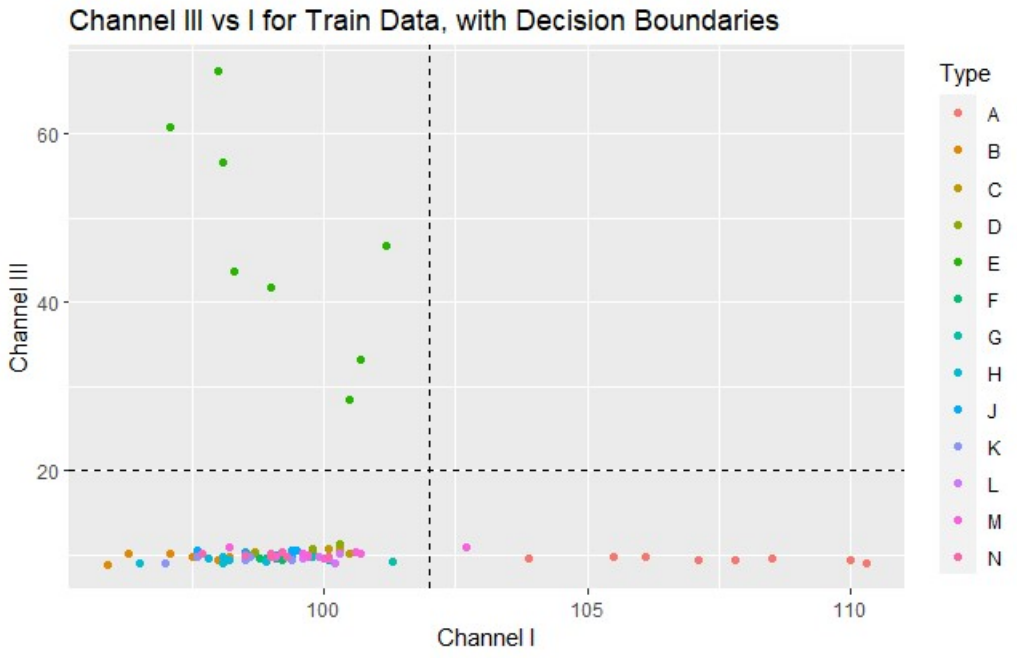

Figure 5: Example of unique classes and their classification

Classes A and E are very different from other classes. Hence they are not being combined. Classification to class A or E is based on a manual decision boundary. In the above Figure, the dotted line represents $I = 102$ and $III = 20$, which is the decision boundary. Observation would be classified as A if $I > 102$ and as E if $III > 20$. There is one observation of class M that would be classified as A. Such observation has been classified as an outlier (by analyzing the boxplot) and hence, should not be considered in creating a decision boundary. After classification has been carried out, observations with predicted class A and E are removed.

The above classes were too distinct to combine them with any other class. An example of a combined class is cluster D, J, and M, visualized in Figure 6.

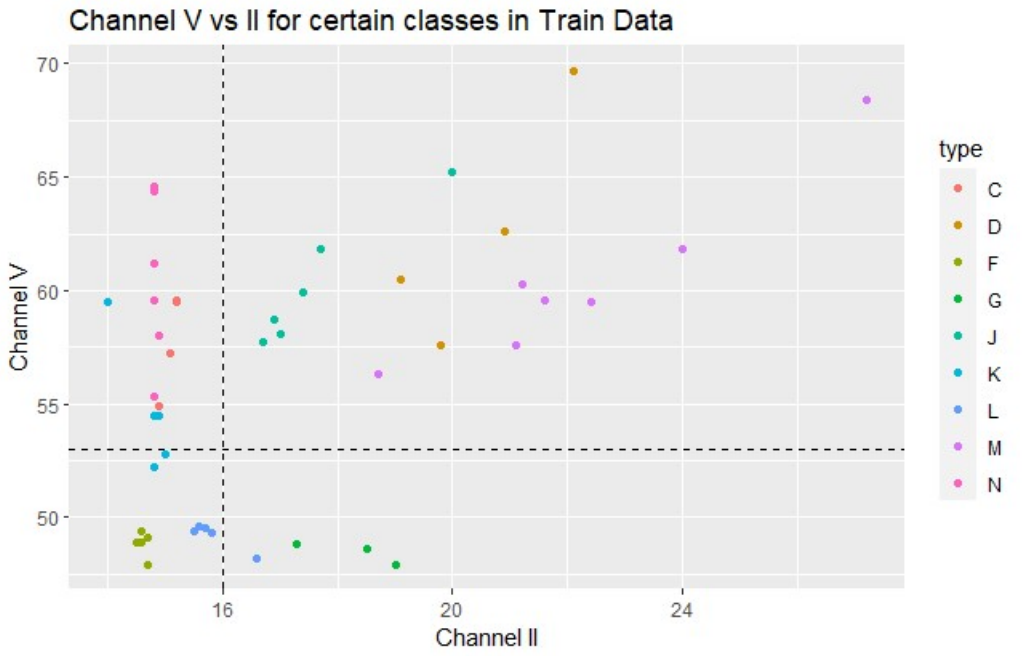

Figure 6: Example of a combined class

Observation is either of class D, J, or M if it satisfies $II > 16$ and $V > 53$. By looking at the above plot, one can see that these classes are structured similarly. Hence, it makes sense to combine them. This will decrease overfit and regression errors due to misclassification. Suppose an observation is misclassified as N (red), while it is J. In such a case, this observation will be fitted to a completely different regression model, creating a significant error. Combining classes reduces the probability of that happening. To differentiate (classify) between classes Linear Discriminant Analysis (LDA) is used.

The above two examples demonstrate classification for a single class and a cluster of classes. The methodology is being repeated for all other classes. The summary of classification can be found in Figure 7.

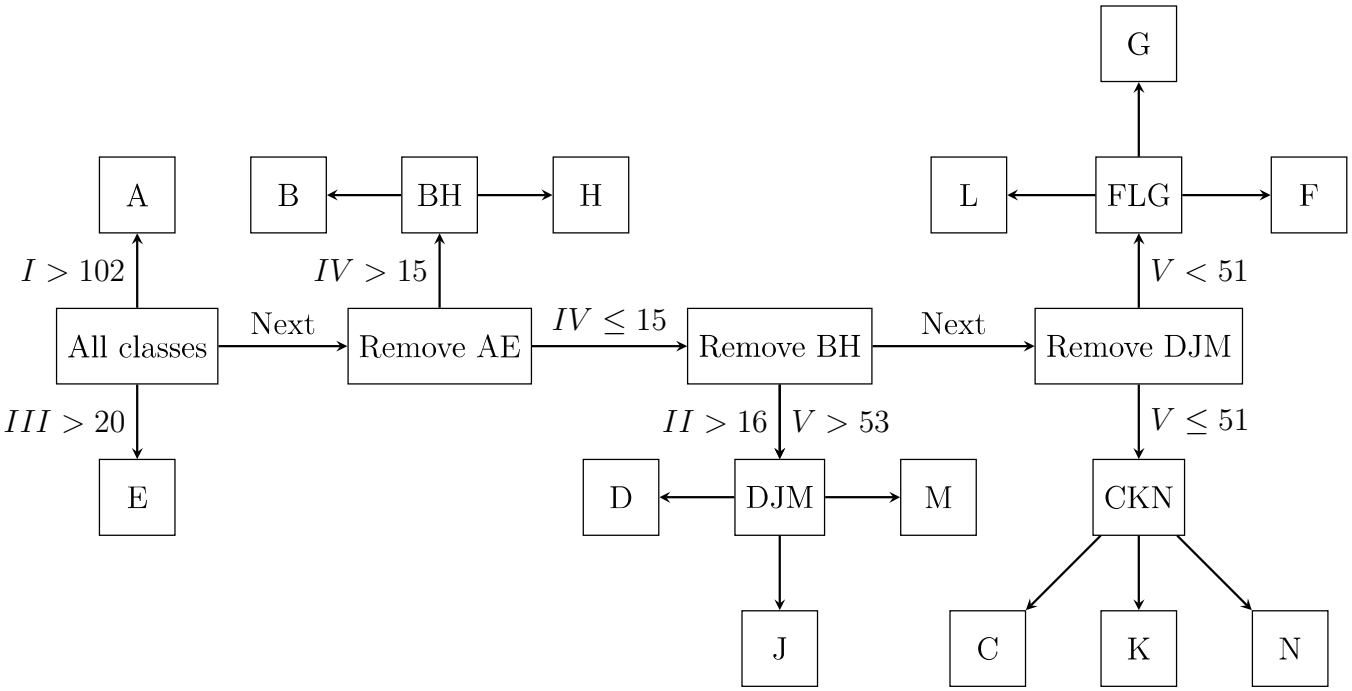

Figure 7: Classification structure

In the above figure, arrows with no label denote the decision boundary, as found by LDA. Arrows with a region (such as $I > 102$) denote manual boundary. Classes A and E are the only distinct classes. Joined clusters include BH, DJM, FLG and CKN. As explained in section 3, observations are classified as pure, if the temperature variable is below certain threshold. In this case, it is $Temp < 24.5$. For a visual representation, see Figure 4 in section 3.

The above approach has some limitations. Determining which classes are similar is based on visual assessment. For fitted model A, there is only an intercept, which can be improved. As described in section 3, the problem of pure classes is very difficult to solve. This approach looks sensible but has no theoretical justification. In addition, some decision boundaries are manual and hence, very subjective. To improve on these issues, classification can be carried out by combining strengths from initial and intermediate modelling.

4.1.3 Final Modelling

The classification for the final model combines the advantages of both, LDA and Logistic Regression to improve the performance of the model. From 6 features, obtained after LDA, only 5 were found to contribute, as determined by projecting the features into eigenvectors. This is being summarised in Figure 8.

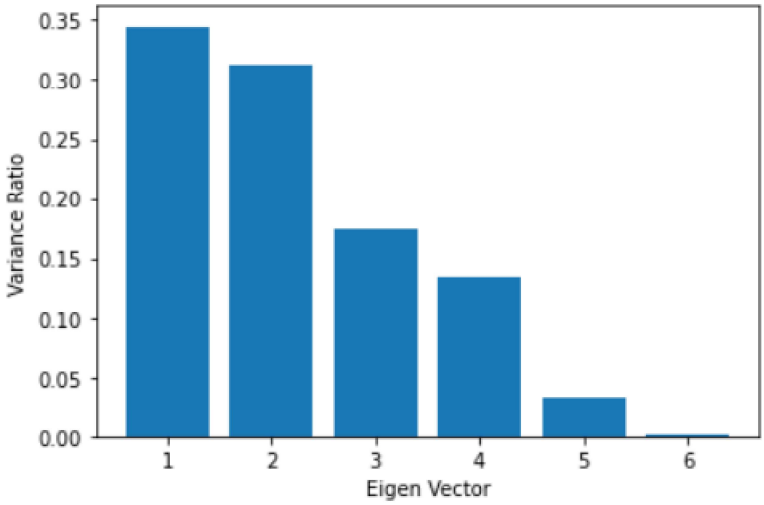

Figure 8: Variability captured by Eigen Vectors

Hence, LDA with 5 components is used first. It tries to find the linear combinations of features that best differentiate between the different classes. The main reason for using LDA first is to project the data onto a lower-dimensional subspace, maximizing the separation between classes while minimizing variation within each class. Further, Logistic Regression uses the reduced-dimensionality features obtained from LDA to fit the model. The plot of eigen-vectors is displayed in Figure 9.

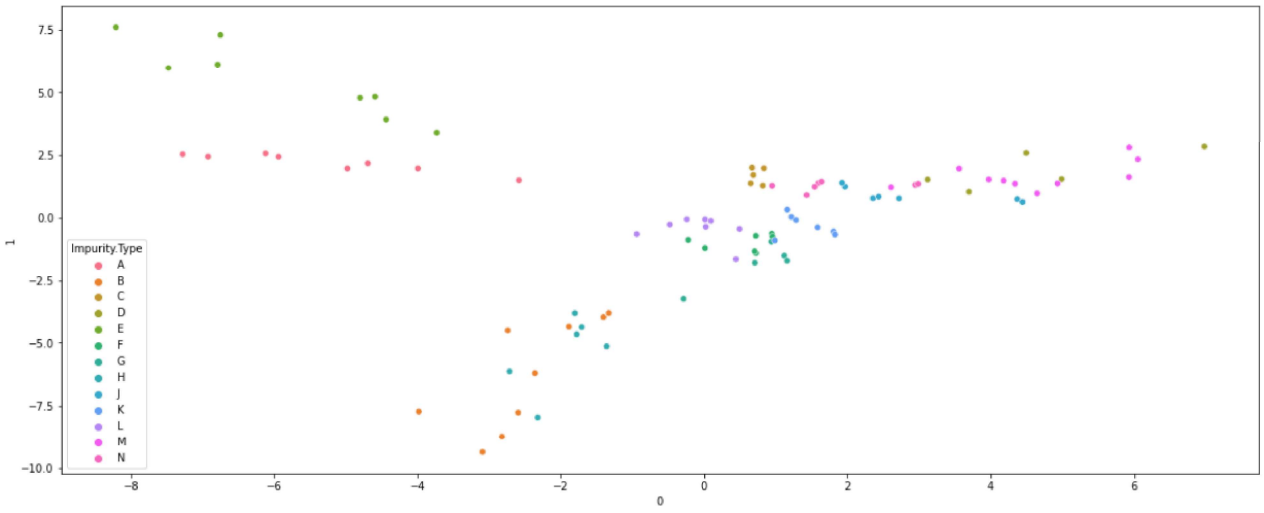

Figure 9: Plot of first two Eigen Vectors after LDA

These features represent the data points projected onto the most discriminative directions identified by LDA. As a result of this modeling, the classification accuracy improved to 0.915 ± 0.103 which was validated using 10-fold cross-validation. The performance of the model improved, because LDA reduced the number of features, potentially improving model training efficiency and thereby reducing the risk of overfitting. Based on this model and with the help of regression, 9 observations have been classified as pure. For more details, see section 4.2.

4.2 Regression

4.2.1 Initial Modelling

EDA showed a linear relationship between percent impurity and features. The baseline model uses this observation to fit a linear regression model with Channels I-V and Temperature as independent variables. The model can be described by the following equation.

$$y_i = \beta_0 + \sum_{j=1}^{6} \beta_j x_{ij} + \epsilon_i \tag{4.2.1}$$

The validation error due to regression is around 81.95 which is high when compared to more advanced models discussed later on. The metric R^2 shows that only 40.74% of the variability in independent variables can be explained by dependent variables. Primary reasons for underfitting include: linear regression being susceptible to outliers, there is a weak linear correlation between different features, and residuals not being independent. This fact can be checked, using Figure 10.

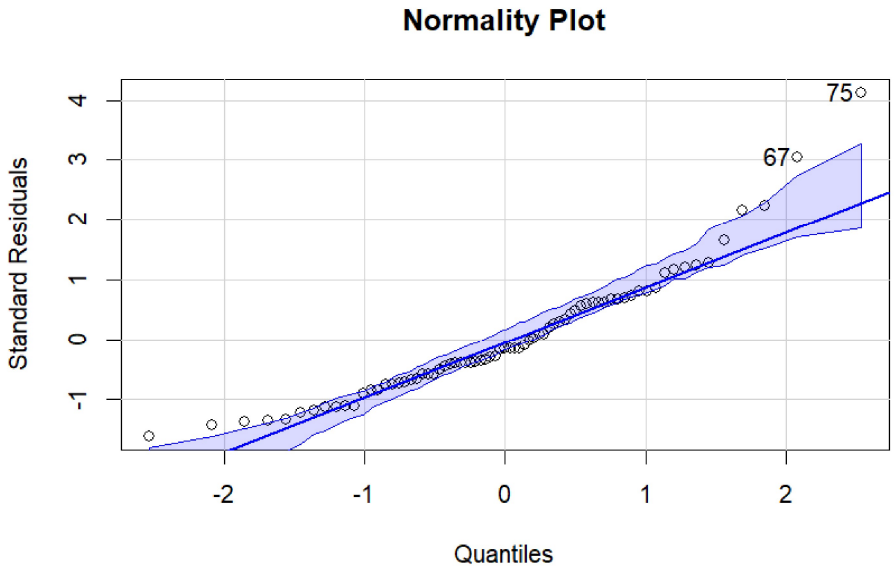

Figure 10: Normality Plot

Clearly, values near the edges deviate from the straight line. This suggests non-independence. Residuals not being homogenous and independent is not only evident from the normality plot (Figure 10) but also from the Shapiro-Wilk test. The following Figure11 helps in explaining the non-linear trend.

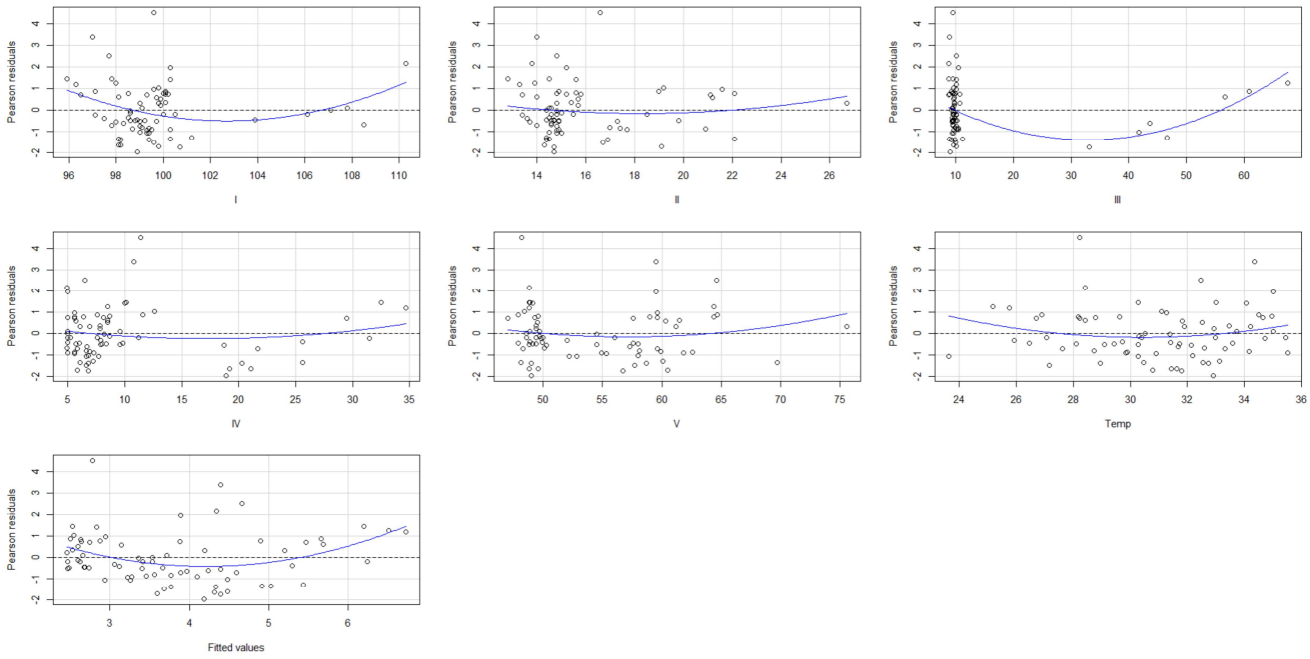

Figure 11: Residual Plots

These graphs show that certain fitted values are not randomly scattered. This, combined with a low p-value for the Tukey test may indicate a higher order of association between dependent and independent features. Secondly, by ignoring the multi-level data structure, the independence assumption might not have been met.

Since this approach does not include inter-class correlation, the next logical progression is to regress, based on classes.

4.2.2 Intermediate Modelling

The intermediate model aims to improve on the issues from the initial model, as well as to apply observations from EDA. The overview of the modeling is as follows.

- Treat classification and regression problems separately. Start with classification, then use regression based on classes

- Join together similar classes. If a class is significantly different from others, do not join it
- Models are fit, based on different classes. To remove redundant features, Lasso is being used

Lasso is a regularization technique, similar to Ridge considered in lecture notes. In addition to shrinking parameters, it can also perform a feature selection (see pages 244-245 of [5]). The penalty function takes the form (4.2.2), where $\lambda \geq 0$ is the penalty term (see page 242 of [5]).

$$\lambda \sum_{j=1}^{6} |\beta_j| \tag{4.2.2}$$

Note that the intercept is not penalized. In contrast to Ridge, there is no explicit solution to Lasso, due to the non-differentiable nature of the modulus function. However, approximation can be found using numerical methods. Package *glmnet* finds the approximation automatically. There are 6 separate models, corresponding to each class/cluster of classes. The penalty term has been found, using a 10-fold CV, by the function *cv.glmnet*. The fitted model for each class/cluster of classes Q is (4.2.3).

$$\hat{y}_i^Q = \hat{\beta}_0^Q + \sum_{j=1}^{6} \hat{\beta}_j^Q x_{ij} \tag{4.2.3}$$

Estimated coefficient for each of 6 models can be found in Table 6. Note that values have been rounded to 2 decimal places.

Fitted Model	$\hat{\beta}_0$	$\hat{\beta}_1$	$\hat{\beta}_2$	$\hat{\beta}_3$	$\hat{\beta}_4$	$\hat{\beta}_5$	$\hat{\beta}_6$
$\hat{y}_i^A$	3.72	×	×	×	×	×	×
$\hat{y}_i^E$	10.61	-0.14	×	0.11	0.44	×	×
$\hat{y}_i^{BH}$	30.35	-0.09	-1.44	×	0.17	×	-0.05
$\hat{y}_i^{DJM}$	-6.33	-0.04	0.50	-0.21	1.04	-0.01	0.01
$\hat{y}_i^{FLG}$	3.42	×	×	-0.57	0.39	×	0.05
$\hat{y}_i^{CKN}$	-16.30	×	×	×	0.24	0.30	0.05

Table 6: Estimated parameters for each class/cluster of classes

Here, × denotes the coefficient 0. Such parameters were identified as not significant, and hence, Lasso shrinks them to 0. For cluster DJM, Lasso did not identify any redundant features. However, for class A, all features were found to be insignificant. Therefore, only the intercept term remains (which is the mean of responses). The impurity percent for chemicals that were estimated to be pure, has been set to 0. The above approach leads to the Validation error, due to regression of 22.41.

4.2.3 Final Modelling

Earlier two sections mentioned the linear models and their shortcomings in predicting pure classes. The main reason is that observations were assumed to be independent. This is not the case, there exist different clusters within. This led us to study Intra Class Correlation which is defined as the proportion of variation in the outcome's between-cluster variance versus the total variation present in the data [6]. Due to its nested structure, the data within clusters are correlated and, therefore, dependent. If the multi-level nature of the data is ignored and a classical linear regression is used to analyze the correlated data, the standard error of the parameter estimates will be wrong. As can be seen from the plot 12 below, the intercepts and slopes are different for each class. This motivated us to use

random-intercept and random-slope multi-level modeling (MLM).

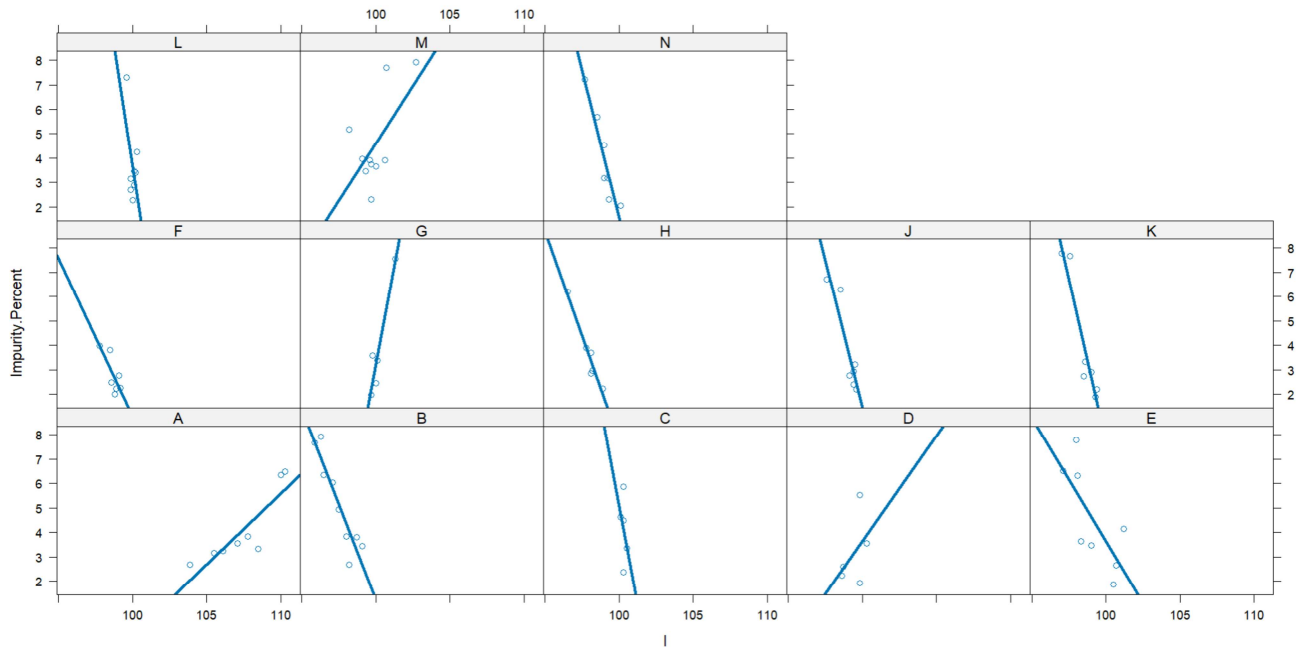

Figure 12: Fitting Linear model for each class

The models being fitted can be explained as follows [7].

$$Level\ \ 1: \ y_i = \beta_0 + \sum_{j=1}^{6} \beta_j x_{ij} + \epsilon_i$$

$$\begin{aligned} Level\ \ 2: \beta_{0_j} &= \beta_0 + U_{0_j} \\ \beta_{1_j} &= \beta_1 + U_{1_j} \\ \beta_{2_j} &= \beta_2 + U_{2_j} \\ \beta_{3_j} &= \beta_3 + U_{3_j} \\ \beta_{4_j} &= \beta_4 + U_{4_j} \\ \beta_{5_j} &= \beta_5 + U_{5_j} \\ \beta_{6_j} &= \beta_6 + U_{6_j} \end{aligned}$$

In this model, β_0 represents the grant mean/ average or general intercept value that holds across clusters. It is fixed across all clusters. The notation U_{0j} denotes a cluster-specific effect on the intercept for channel I. Similarly, to incorporate the different impacts between clusters, the assumption is that the cluster-specific slope β_{1j} changes from cluster to cluster as a random effect β_1+U_{1j}(for channel I), with a mean of β_1. $U_{ij}$is a random effect because it varies from cluster to cluster with a mean of 0 and a variance τ^2. The error term ϵ_{ij} denotes the within-cluster variation with a within-cluster variance of σ^2. Note a different notation, it is not a parameter. MLM further assumes that U_{ij} and ϵ_{ij} are uncorrelated. EDA found that data is positively skewed. Hence, a square root transformation on channels I, II, IV, and V is helpful to normalize the distribution. It leads to a more accurate representation of the relationship between the variables.

The validation error due to regression decreased to 16.50, which shows an improvement in the model. However, it also led to the problem of overfitting. Some predicted values were abnormally high. It can be concluded that one should look for nonlinear options.

From the residual plot, Figure 11 one can infer that residuals are not independent, especially for channels I and IV. So the focus has been shifted from General Linear Models (GLM) to the Generalised Additive Models (GAMs). GAMs provide a general framework for extending standard linear models by allowing non-linear functions of each of the variables while maintaining additivity. The nonlinear relationship between each feature and the response is to replace each linear component $\beta_j x_{ij}$ with a smooth non-linear function $f_j(x_{ij})$[8]. The model can be written as (4.2.4).

$$y_i = \beta_0 + \sum_{j=1}^{6} f_j(x_{ij}) + \epsilon_i \tag{4.2.4}$$

The following figure shows the degree of Splines found.

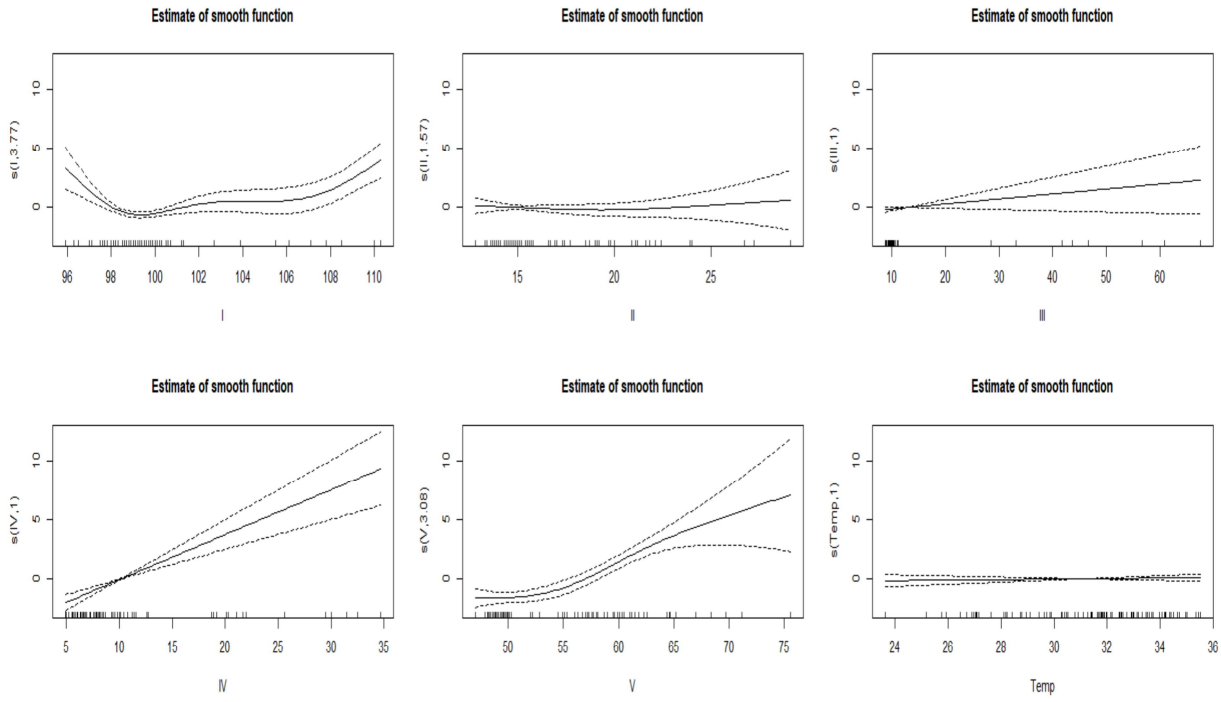

Figure 13: Estimated Smooths for different features

This figure shows that some variables are linear, while others are not. Since the data is split into 80:20 ratio, there are only 73 instances for training GAM. This is very low to reliably capture non-linearity in the data set. So for the final week prediction, an ensemble model of GLM and GAM is used. GLMs capture linear relationships effectively, while GAMs can handle non-linear and complex interactions between variables. Combining them allows the ensemble to capture a wider range of relationships in the data, potentially leading to better predictions on unseen data. GLMs act as a regularizer, preventing the ensemble from adapting too closely to the training data's noise. The goodness of fit can be checked with the help of the following plot.

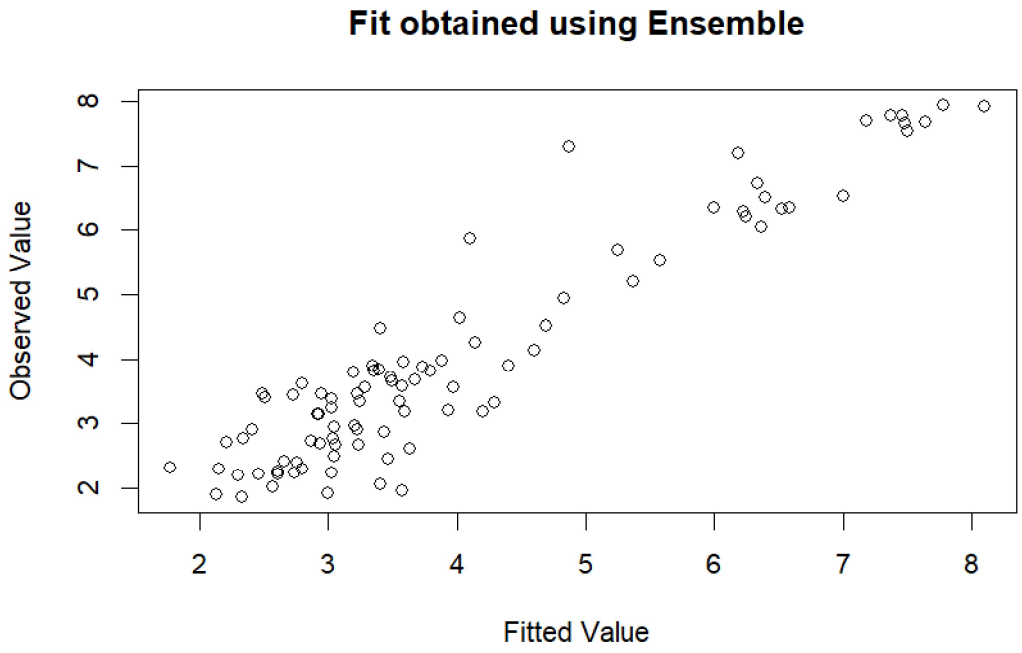

Figure 14: Fit obtained using Ensemble

The plot above compares values predicted by the model with actual values for training data. A linear trend in the plot and all the points being centered around the 45-degree line show that predicted values are almost equal to actual values. The following table shows the change in Validation error for classification and regression.

Table 7: Classification Validation Error

Classification Method	Validation Error
Initial	0.83 (±0.103)
Intermediate	0.89
Final	0.915 (±0.103)

Table 8: Regression Validation Error

Regression Method	Validation Error
Initial	81.95
Intermediate	22.41
Final	16.50

The validation error for both, classification and regression has been steadily decreasing

over the weeks. The discussion for Test score (prediction data) is given in the next section. Note that conclusion is given after Part II.

4.3 Discussion

The aim of Part I was to accurately classify impure and pure chemicals, as well as predict their concentration. The initial model used basic modeling to achieve a good initial score (on Test data) of 323. The main problems with this approach include fitting with redundant features, not classifying any pure chemicals, and (naively) assuming the independence of residuals. Classification was done using logistic regression. Second's week approach included using Neighbourhood component analysis and KNN for classification and Random Forest Regressor for regression (not discussed in section 4). This led to a bad score of 1293 so the idea was dropped. The reasons for this include poor performance of KNN as K-NN is sensitive to the local structure of the data (skewness is a bit different for two sets), and also small value of K (neighbors) led to overfitting. Third's week model (the Intermediate Model) corrected previous weeks' mistakes, by using more advanced, yet easily interpretable machine learning techniques. This approach led to an improved score of 258. Relying on manual clustering of pure classes led to misclassification. Our inability to identify pure classes through regression led us to explore the nonlinear model GAM. It helped in the identification of 7 instances as pure class. This approach has further improved the score to 155. The final model improved the score to 139, due to using an ensemble technique, combining GAM and GLM. Overall, the score has been decreasing from week to week (with one exception), indicating logical model development. Interestingly, the score has improved, by a large margin in the initial weeks and the rate of improvement decreases in further weeks (as expected). Simple models performed surprisingly well, while more complicated methods improved on the Test score.