0.1 Description of the Problem

Classify the superficial faults of stainless steel plates, that may be generated in the production phase of the plates. There are 7 possible faults to be recognised. These are indicated as:

1. Pastry;
2. Z-scratch;
3. K-scratch;
4. Stains;
5. Dirtiness;
6. Bumps;
7. Other faults.

The classification task is to be obtained using 27 physical and appearance features.

0.2 Exploratory Data Analysis

The dataset is composed of 1941 observations and 27 variables. 25 out of the 27 variables are numerical and the remaining two categorical, describing the type of steel used in the plates. The faults are distributed as seen in Table 1. It can be seen that the fault classes are not balanced. This will definitely have an influence on the quality of the predictions. For instance, it can be expected that the classes with lower numerosity will have a higher misclassification - or other type of errors - rate. There are no missing values in the dataset.

fault type	Pastry	Z-Scratch	K-Scratch	Stains	Dirtiness	Bumps	Other faults
numerosity	158	190	391	72	55	402	673

Table 1: Numerosity of fault types.

The first step in the analysis of the problem is exploring the dataset. A first glance of the histogram of the variables is available at `https://bigml.com/user/czuriaga/gallery/dataset/50b8c55a035d07198d00007c`. Here, only a selection of histograms and scatterplots will be shown. Figure 1 shows four examples of histograms for 4 selected features. Histograms give a first idea of how the data are distributed. However, they are only *univariate* instruments, meaning that they treat the variables independently. To have a better understanding of how the features relate to one another, *scatterplots* can be used. Figure 2, for instance, shows the scatterplot between Minimum and Maximum Of Luminosity. Each dot in the plot corresponds to a pair (x, y), with $x =$Minimum of

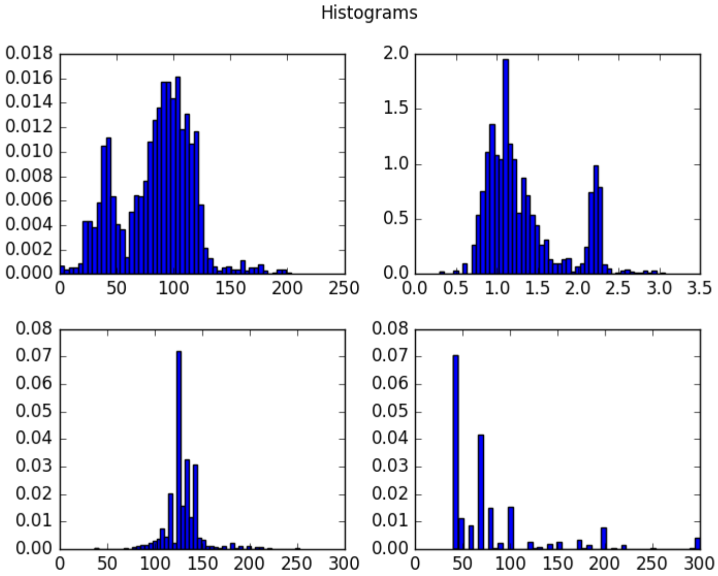

Figure 1: Examples of histograms for some selected variables. Top left: Minimum of Luminosity, Bottom Left: Maximum Of Luminosity, Top Right: Log X Index, Bottom Right: Steel Plate Thickness.

Luminosity and $y =$Maximum of Luminosity, for an observation in the dataset. Ideally, the *perfect separation case* would be to observe, in each scatterplot, seven differently coloured clusters of points. Such a configuration would mean that the faults show different distributions among their variables. Plotting all the scatterplots is a heavy task. Given n variables, there are $(n^2 - n)/2$ distinct plots; in this problems, it means 351 of them. Another utility of scatterplots, besides shoeing the correlation structure, is that they may suggest an idea of the direction to follow to approach the problem.

Other ways to investigate how the features behave, is to look at the covariance matrix for each of the fault types or to perform a *Principal Components Analysis* (PCA). The covariance matrix is such that the element (i, j) represents the covariance between the $i-$th and $j-$th variable. PCA, instead, is a technique that achieves simoultaneously two objectives: on one hand it allows for a dimensionality reduction - in terms of final number of meaningful variables to consider - of the problem. On the other hand, it finds the best linear combination of variables (principal components) explaining the distribution of variance in the data. In the problem at hand, the PCA brought inconclusive results. By dividing the data into fault types, it appeared that the principal components were all very similar among the groups. Therefore, not much information to discriminate different fault types can be retained from this analysis.

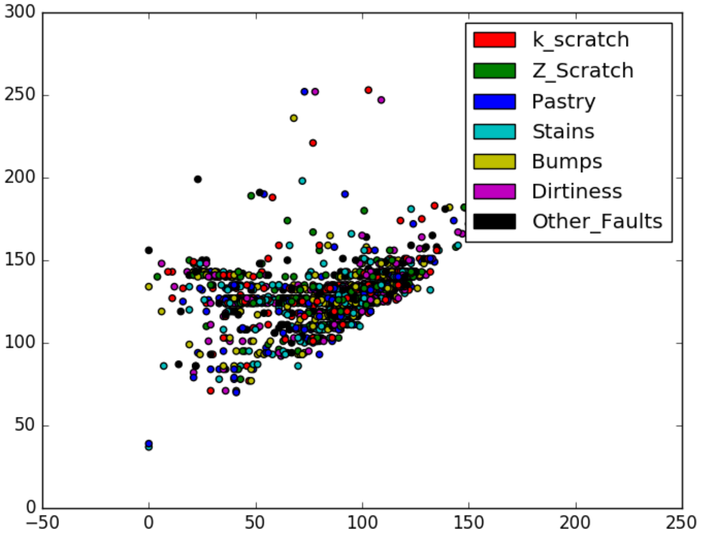

Figure 2: Example of scatterplot between Minimum Of Luminosity and Maximum Of Luminosity.

0.3 Data Modelling

The complex structure of the scatterplots makes unlikely that methods like Naive Bayes Classification or Support Vector Machine could produce good results for this specific problem. Therefore, different methods, namely Random Forests (RF) and Gradient Boosting (GB) Classifiers, were tried out. These methods are more complex than the first two cited; however, they are simpler than other bleeding edge methods coming from the field of *deep learning*. Moreover, RF and GB are suitable methods to handle mixed variable types - numeric and categorical - and that is the reason why they were first chosen for this problem.

The first operation to perform is to divide the dataset into a *training set* and a *testing set*. As the names suggest, the training set will be used to learn the statistical model that will be used to classify the observation. The testing set will be used to test the model built. This is a necessary operation: if a model were built and tested on the same set of data, the error committed would be highly underestimated.

Assuming the data are all independent, in order to build a training set, from each of the fault classes, an 80% of the observations was randomly sampled. The sampled 80% of the whole dataset was retained as training set, whereas the remaining 20% constituted the testing set.

As previously stated, Random Forests and Gradient Boosting Classifiers were learnt on the training set. Some *hyperparameters* need to be set in these methods, but the optimal values are not known. Usually, a *grid searching* approach is adopted to set the values of those parameters. The same procedure was used in this problem. Grid searching consists in computing the classifier for each given subset of the parameters. At the end, the classifier with the lowest error - in some sense depending on the method - is retained.
Using the described procedure, an optimal Random Forests and an optimal Gradient Boosting Classifier were built.
Better methods may also be produced by interfacing with the engineering teams to understand that some faults may be more serious than others and therefore it could be more important to detect those ones, in order to achieve a more relevant classification. This is usually done by setting different misclassification costs in the loss functions of the methods.

0.4 Results and Further Comments

The models were evaluated on the testing set. The evaluation was performed using two measures:

1. mean squared error (MSE);

2. misclassification error.

The former gives a global evaluation of the model, the latter a more class-wise performance. The MSE is simply the mean, among the classes, of the squared misclassification error. Table 2 shows the MSE values for the two methods. It can be seen that the Gradient Boosting performs slightly better than RF in terms of MSE.

	RF	GBM
MSE	3.0265	2.733

Table 2: MSE for the two trained methods.

The misclassification error is checked by looking at the *confusion matrix*. The confusion matrix is a matrix such that the rows represent instances of the predicted class and columns of the true one. Therefore, the element (i, j) of the matrix corresponds to the number of cases truly belonging to class i that were classified as coming from class j.

In a classification task it is very unlikely that all the features contribute in the same way to the goodness of the method. Random Forests and Gradient Boosting compute also a measure of importance for each of the features used. This information can be used to draw a *partial dependence plot*. A partial dependence plot shows the dependence between the target function of the classifier and a set of target features, marginalizing over the values of all other variables.

Predicted \ True	Pastry	Z-Scratch	K-Scratch	Stains	Dirtiness	Bumps	Other faults
Pastry	34	0	1	0	2	4	12
Z-Scratch	0	62	0	0	0	0	6
K-Scratch	0	1	171	0	0	0	2
Stains	0	0	0	27	0	1	2
Dirtiness	0	0	0	0	17	1	0
Bumps	3	2	2	1	1	105	45
Other Faults	29	17	6	3	5	66	240

Table 3: Confusion matrix for the Random Forests classifier.

Predicted \ True	Pastry	Z-Scratch	K-Scratch	Stains	Dirtiness	Bumps	Other faults
Pastry	38	0	0	0	2	7	19
Z-Scratch	1	74	0	0	0	0	4
K-Scratch	0	2	171	0	0	0	5
Stains	0	0	1	26	0	1	0
Dirtiness	0	0	0	0	15	0	0
Bumps	11	0	0	1	2	115	45
Other Faults	16	6	8	4	6	54	234

Table 4: Confusion matrix for the Gradient Boosted classifier.

From Table 3 and Table 4, it can be seen that the Gradient Boosting is able to recognize Z-scratch and Bumps faults better than the Random Forests and this is probably the reason for the Gradient Boosting's lower MSE. The most critical classes for both methods are Bumps and Other Faults. In case of the Other Faults class, a possible explanation may be that this group is a container group for many not well defined faults that share many similarities with the other flaws. A way to achieve better results could be to further investigate the Other Faults class, by looking at possible clusters that may define sub-types of faults.

Other possible ways to have better classifications may be a more exhaustive grid search for parameters optimisation or considering more complex relations between the variables used.

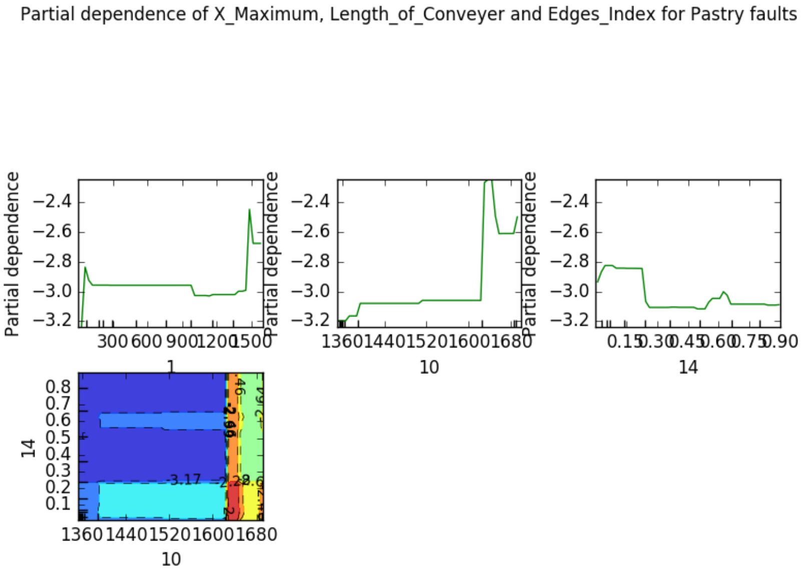

Figure 3: Example of partial dependence plot (PDP) for the Gradient Boosting. From Left to right: PDP of X-Maximum, PDP of Length of Conveyer and PDP of Edges Index for the Pastry faults. Those features correspond to the three most important for the Gradient Boosting. The bottom-left plot shows the partial dependence of the target function when Length of Conveyer and Edges Index are taken into account simultaneously.