# Hella scrap modelling

Authors: Andrej Borštnik, Luka Stopar

The task of this project was to model scrapped lamps produced by Hella. We were provided with diverse data, from different sources, with different structures, with different time scale, and of different quality. A problem is that the target variable is less a lot less frequent than other data and therefore even a lot of data (overall) is actually very little data on which we can optimize our model.

## Handling and processing the data

The first challenge was aggregating all of the data into feature vectors, so that we could recognize what contributes to the scrap. There are different scrap types that are probably caused by different factors. We model the scrap by using several different machine learning models, which we will present later. But first let us look at the data structure.

Firstly we had machine data (temperatures, velocity of parts, etc.), sampled approximately every minute. We also had some environmental data (dust particles), that was expected to influence some scrap types (mostly connected to lacquer application). This data was in .csv format and was easily handled.

Next we had a daily work plan in the form of a .pptx presentation. Here details about which products go onto which machine and at which time were provided. Extraction was however not that simple, as times of product changes were written in the miscellanies text, and mostly need human intervention to be perfectly re-constructible. Moreover the product names are not standardized here and hence matching product names to real products is not completely reliable.

Lastly we were provided with scrap numbers, arranged by product, in .xslx format. The naming problem from previous data continues to reappear here (even though names are standardized here), as we cannot perfectly match previous product to their scrap values. Moreover scrap values are collected only at the end of each shift, making it hard to correlate them to any machine values that are collected continuously.

We modelled dust particles separately from other variables, as they only influenced certain types of scrap and are unlikely to contribute to the same scrap that machine variables do. The function we used was in essence the same as for machine values, as we only copy it from its .csv file. Hence we will only describe how we transformed data with machine values.

As we described above the data is provided from three different sources and therefore we wrote three different functions, to handle each of them. For convenience we used the expressive *python* programming language.

We saved the .pptx presentation in .xml format and extracted the table and miscellanies text with the *parseXmlFiles* method. This method takes parameters that describe which files should be parsed and returns their dictionary representation in python.

In Figure 1 we can see the daily plan for 1.10.2015 and in Figure 2 we can see the python (simplified) representation of the data.



Figure 1: An example of a daily plan.

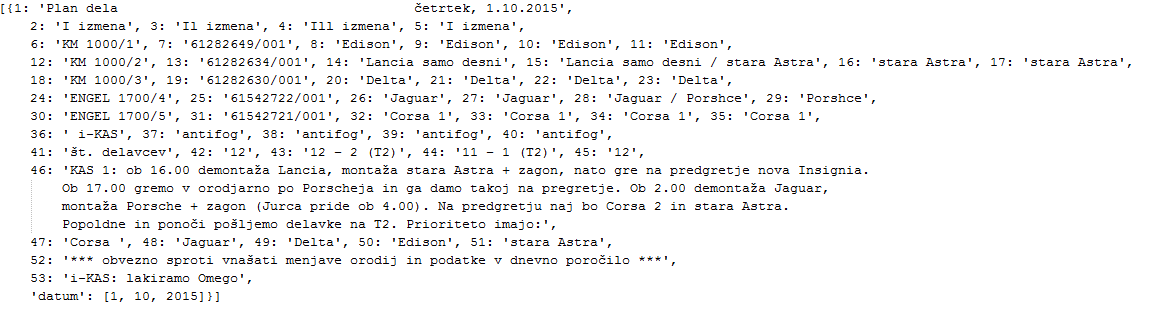


Figure 2: An example of a daily plan in python representation.

We do a similar thing for the excel file, that we save as .csv file, using the *parseCsvFiles* method. And in the *parse* we parse the .csv machine value file and combine it with results from the other parsed data. This method outputs the results in a specified .csv format file and returns it, sorted/collected by shift, as our primary variable is only recorded every shift.

When we output the .csv file we cannot collect the data by shift (as we need feature vectors for every timestamp we use) and must therefore improvise. Currently we just write the entire scrap of the shift for each timestamp. This way only the scale of the value is distorted and no information is lost.

## Modelling

We tried to apply many different models, with different approaches to handle data, but in the end could extract very little new information form the dataset.

The first model we tried is mathematically the most “correct”, as we used logistical regression, where we evaluated our target variable only once per shift. To be more precise we built a model, that assigns a weight to each variable, but the function we optimize is not computed on each timestamp, but rather on all timestamps belonging to a shift. We used the Newton method to find a (local) minimum of the optimization function. This way we exactly follow the data structure and therefore we expected to get the best results. But that was not the case. The results were not good in either model, but they were a bit better in the other models.

Than we tried to model the scrap straight from the feature vectors (disregarding the data restrictions) and also from averaged feature vectors – on different time intervals. With the averaging we expected to lose the finer aspects of the model – and therefore precision, but in turn emphasizing the possible single underlying cause of a scrap type.

This method provided more success than the first one, both in actual precision – the total feature vector space - and in identifying the underlying cause.

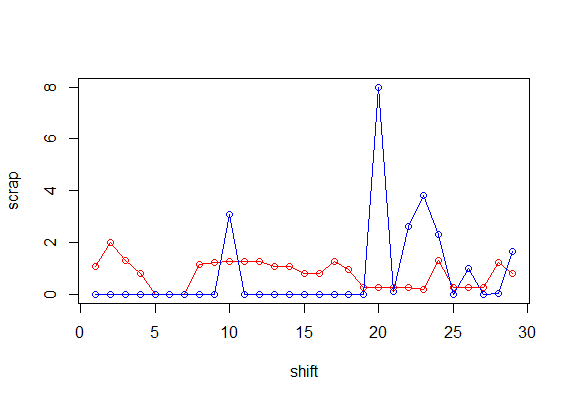


Figure 3: Red prediction and blue the actual scrap.

Because of the vast dataset we mostly used linear models – the linear and logistic regression. We also tried other models on smaller (averaged) datasets, but the results were worse, that with a linear model.

We built the linear models with this procedure:

* Build the model on the entire dataset.
* Identify the significant variables and discard the insignificant ones, so that we do not over fit our model.
* Build the model again using only the selected variables.
* Evaluate the model.

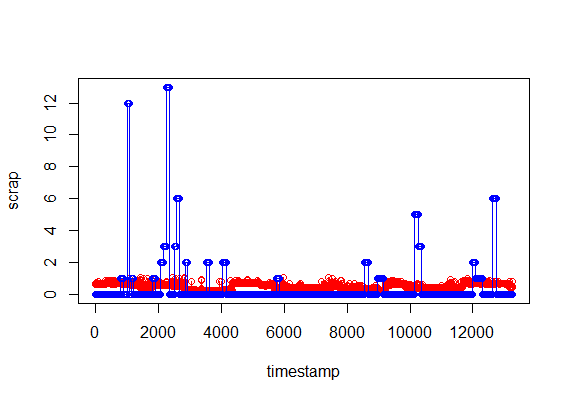
All the models we ran had RMSE (root mean square error) variance similar to the actual (computed) RMSE. From this alone we can conclude we extracted almost no information from the dataset, as – statistically – just predicting the average scrap would yield results of similar quality. 

Figure 4: Red is predicted and blue the actual scrap.

The only significant thing we could gather is that a certain variable is correlated to scrap by an order of magnitude more than any other collected variable. This variable is **Čas(pavze Z)[s]** - that is the time of pause of some machine element. Why this is an important variable is up to the engineers to explain – and possibly try to decrease its influence on scrap.