**Klarna Machine Learning Case Study – Reasoning and solution**

1. Data preparation:

The first insights over the data (Done in the file “DataAnalysis/DataAnalysis.py” and summarized in “DataAnalysis/report.html”) stated that many variables were having issues of plenty zeros and null values. Taking this into account, I decided to turn every variable into categorical: Turning null values into a ‘NoData’ category and, for the variables that mainly have zeros, a ‘0’ category. Regarding the numeric variables, I decided to create equiprobable intervals in order to not group much data into one category.

Secondly, I developed an analysis over the response variable. This analysis could be found in the “Graph\_SumDefault” folder. Here, I made 1 graph per variable in order to analyze which is the effect of each category of the variable over the response variable. This analysis summed up the number of defaults per category (Viewed in red) and compared it against the frequency of that category (Viewed in blue).

With this analysis I was able to identify which variables truly made a significative explanation of the default risk by comparing the amount of defaults against the frequency of the category: if the sum of defaults was correlated with the frequency of each category then the variable will have no effect, since the distribution of the response variable is not affected by any category. Thus, if there’s a category that has a significative difference between these distributions, the variable could be important.

1. Model analysis and testing

In the folder “ModelAnalysis” you may find the file “ModelTesting.py” in which I trained a Random Forest and a Neural Network. These two models were initially built to predict the default risk with a training set that contained the default rate of 1.4%. Naturally, this caused the models to approach the naïve response and return 99% zeros instead of predicting the 1.4% ones.

Thus, I decided to undersample the majority class (aka the zeros) in order to make the models adjust better to the characteristics of the ones. Therefore, I created a new training set with 50% ones and zeros and trained the models with this data.

The two models were evaluated regarding their performance in the metric recall over the test sample (Which contained a default rate of 1.4%). This, considering that it is more important to predict correctly the default risk than the non-default. According to this metric, the Neural Network model exceled in this metric considering that it was considered during its training.

1. Model production

The files “1. ModelTraining.py” and “2. ModelExecution.py” contain the execution of the training of the definitive model and the application of this model over the datapoints that didn’t have the “default” variable set. The final answer was saved in the “Answer.csv” file.