6.867 Machine Learning (Fall 2018)
Prof.: Suvrit Sra
TA: Alex Turner

Youssef Berrada (ID: 911844154)
Georgy Guryev (ID: 924667514)
Sheng Yao (ID: 916855968)

Project Milestone 5: Results and Interpretation

In this milestone, we go over the feature engineering approach we have as well as to build a baseline method such that we compare it to.

Data pre-processing

The data pre-processing mainly involves aggregating different static features of the time-series (mean,std,max,min and etc) and the time-series specific feature (autocorrelation etc) from each passband and each object. The time-series specific features were achieved by adopting $extract\_feature()$ in the tsfresh library. The output of the data pre-processing will generate 150 time-series related features, which would be joined with another 10 meta features.

Baseline Model

We will define our baseline model as to predict the most common class of the training data set and we construct the confusion matrix, as well as some performance measure that we will compare it to. By simply aggregating the frequency of each class, we found that class '90' has the highest frequency in the training data. Using class '90' as our baseline prediction, we have the following performance:

Accuracy Score	Recall Score	F1 Score	Log Loss
0.0210	0.0714	0.0325	32.380

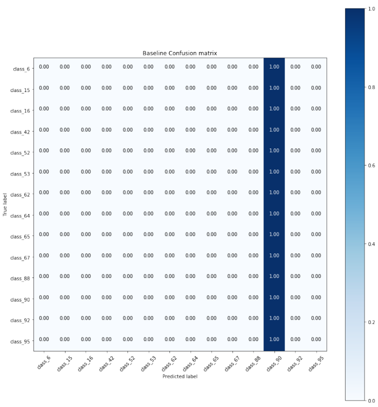
we will use a weighted log loss to assess our results in the same spirit to what Kaggle is doing, such that we have :

$$L = -\frac{\sum_{j=1}^{M} w_j \sum_{i=1}^{N} \frac{1}{N_j} \tau_{i,j} \ln P_{i,j}}{\sum_{j=1}^{M} w_j}$$

However the actual weights of various classes are unknown, therefor we come up with weight approximation based on implicit indicators of class uncertainties. The proposed weights are summarized in the following table:

class	6	15	16	42	52	53	62	64	65	67	88	90	92	95
weights	1	2	1	1	1	1	1	2	1	1	1	1	1	1

The confusion matrix is as follow:



Results and Interpretation

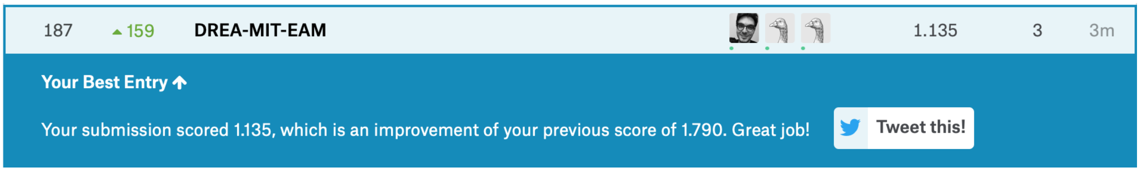
Algorithm	Loss function
Light GBM	0.31
XGBoost	32.15
Decision Tree	5.10
Random Forrest	3.65
Neural Network	27.90

Table 1: Comparison of the resulting loss functions for the implemented ML algorithms

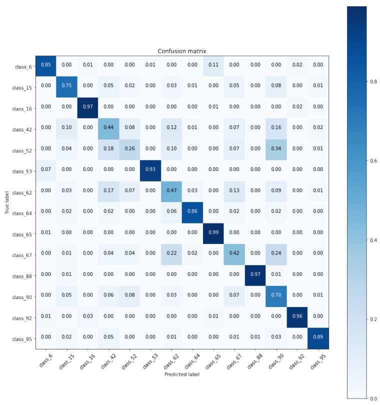
Basically here we observe that the lightgbm model is giving the most promising results at least in sample. We are yet to use the optial tree, where we spend some time setting up to use, so hopefully we will have it for the final delivery. But already, we can see that all decision trees methods overfit pretty quickly on our data, so we need to be careful on how we tune them well. Also for the final delivery, we also decided to split the training data into a train set and test set, to produe out of sample record at least for the class project delivrable.

Kaggle Ranking

After implementing the above, we succeeded in having a score of 1.135 in the Kaggle ranking which led us to become ranked 187^{th}. To achieve that, we selected the Light GBM algorithm that performed well on the test data.



The Confusion Matrix for the Light GBM is the following:



Division of Labor

All three team members contributed significantly to the milestone. We actually all brainstormed at the same time to understand each algorithm and discussed what methods should we explore and package we should use. We then all went over all algorithms and libraries we plan to use.