Energy generation prediction: Lessons learned from the use of Kaggle in Machine Learning Course

Jesus Fernandez-Bes
CIBER-BBN and BSICoS Group
I3A, IIS Aragón, University of Zaragoza
Zaragoza 50018, Spain
jfbes@unizar.es

Jerónimo Arenas-García, Jesús Cid-Sueiro
Signal Theory and Communications Dept.
Universidad Carlos III de Madrid
Leganes 28911, Spain
jarenas@tsc.uc3m.es, jcid@tsc.uc3m.es

Abstract

In this paper we expose the conclusions extracted from the use of a Kaggle Challenge as a tool in teaching a Master level course in Data Processing.

1 Course and Challenge description

The *Data Processing* course of Telecommunication Engineering Master at Universidad Carlos III deals with the principles underlying the regression, classification and other data analysis problems from a practical point of view. Because of that practical orientation, a number of tools were used:

- Ipython notebooks [1] as main lecturing tool.
- A Kaggle challenge as part of the evaluation.

The aim of the challenge was to make students apply the studied methods and principles and to encourage them to explore other techniques. The challenge was hosted in **Kaggle inclass** system, lasted for two months, and meant one third of the final mark for the students, that competed in pairs or individual teams.

The proposed problem was a regression task with 5 variables and missing data. The chosen dataset was the Combined Cycle Power Plant Data Set [2] taken from the UCI repository[1]. In order to make the problem more difficult, and to avoid cheating , as all the labels are available on the internet, the data was corrupted with noise, some instances of variable 1 were removed and an irrelevant variable, uniform noise, was added. The evaluation metric was the root mean square error, RMSE.

2 Evaluation, code submission and cheating prevention

The grading of the challenge consisted of 3 points:

1. 1 point. Kaggle private leaderboard position, up to 1 point for the first team then linearly down to the last team of the classification that obtains 0 points. During competition, students only had access to a public leaderboard calculated on 50% of the test data.
2. 1 point for outperforming a simple linear regression benchmark.
3. 1 point of code quality and methodology evaluation.

The third part of the evaluation was included for two main reasons: The code used to generate the submitted solution was asked to be delivered using the course website, with the aim of preventing cheating as much as possible. In addition, the students were asked to briefly explain their approach. In that way, students that showed a better understanding of their approach could be rewarded.

[1] https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

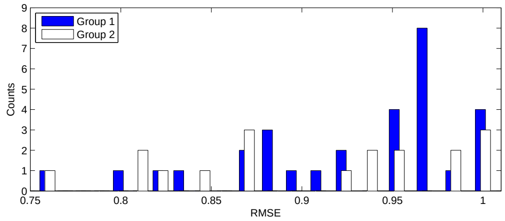

Figure 1: Student relative performance (best Performance / oerformance of the team) histogram

Table 1: Summary of group RMSE (lower is better)

	Group 1 (30 teams)	Group 2 (18 teams)
mean	5.2304	5.7066
std	0.1328	0.5031
min	5.0814	5.0971
max	5.6220	6.8101

3 Student performance

Regarding the performance of the students, the 48 teams participating in the challenge beat the simple benchmark. A relevant fact is that students were divided in two class groups, located at different campuses, and statistically different performances were observed in these two groups. Group 1 obtained a better mean performance, with higher number of submissions and more principled use of methodologies such as cross-validation (according to their description). Group 2 performed less submissions and explored less new techniques.

In fact, most groups used the regression methods presented in the course: linear least squares and Gaussian Processes. However, the best performance was obtained by students who explored new methods, such as ensemble methods. The evolution of the submissions, skipped in this abstract because lack of space shows the evolution of each team very clearly.

In general the performance of the students was satisfactory, being their opinion about the learning process was really positive. This opinion was specially positive in Group 1. Students from Group 1 got more involved in the “game” and became highly motivated.

4 Evaluation process and discussion

As instructors, we feared that there were a lot of cooperation among the teams, ruining the competition. However, although there was some degree of cooperation among the teams, we believe that the competitive element of the challenge and the delivery of the code prevented this to be very harmful. In fact, there were no two groups with the same RMSE, which means that in their final submission they did at least something slightly different.

On the contrary the need to check the scripts of all the groups, to avoid cheating, involved a lot of work to the teachers and limits the applicability of the evaluation method. The use of *Kaggle* online scripting functionality (called *Kaggle kernels*) could solve this problem, provided that they can only be access by the organizers of the challenge. However, this functionality is not implemented yet in *Kaggle inclass* platform.

Overall the Kaggle challenge was considered by the students as their main learning tool. Most of them, got to understand the basic concepts of Machine Learning, such as over-fitting and how to prevent it, when working in the challenge. In conclusion, although there are problems regarding the cheating control, we strongly advice the use of these tools in data processing course.

Acknowledgments

This work was partially supported by projects TEC2014-52289-R, and PRICAM S2013/ICE-2933.

References

[1] Fernando Pérez and Brian E. Granger. IPython: a system for interactive scientific computing. *Computing in Science and Engineering*, 9(3):21–29, May 2007.

[2] Pınar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power & Energy Systems*, 60:126–140, 2014.