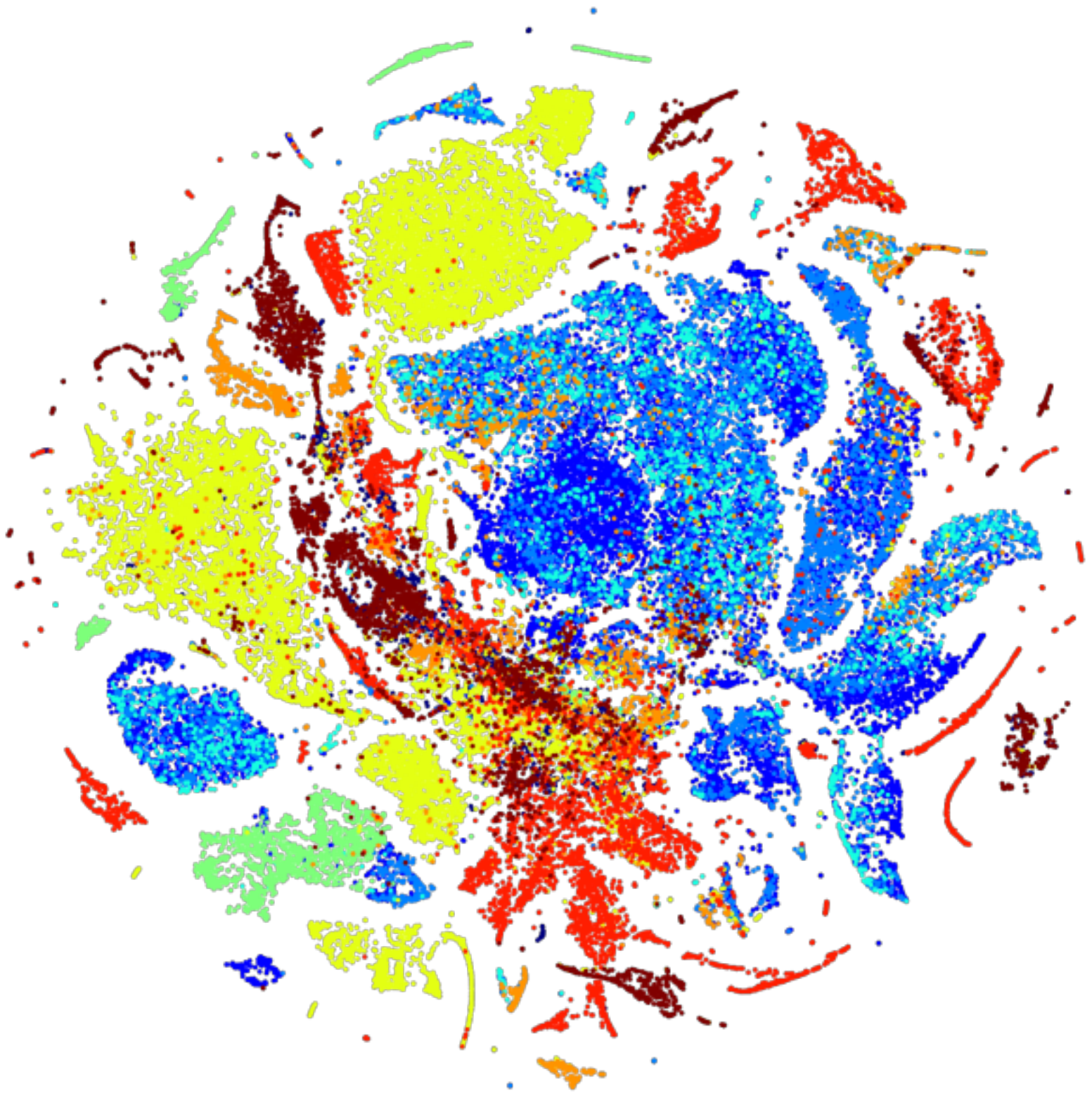
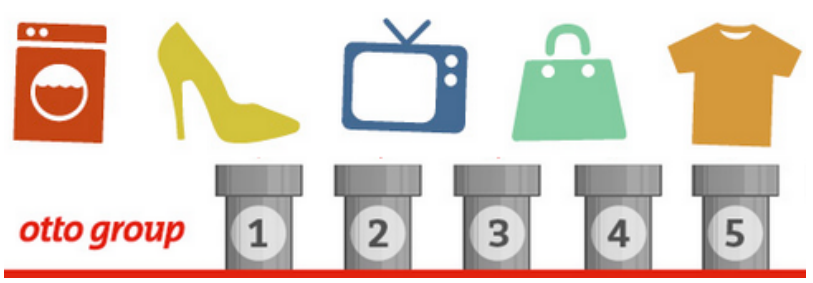
Otto Product Classification Winner's Interview: 2nd place, Alexander Guschin ¯_(ツ)_/¯



The Otto Group Product Classification Challenge made Kaggle history as our most popular competition ever. Alexander Guschin finished in 2nd place ahead of 3,845 other data scientists. In this blog, Alexander shares his stacking centered approach and explains why you should never underestimate the nearest neighbours algorithm.

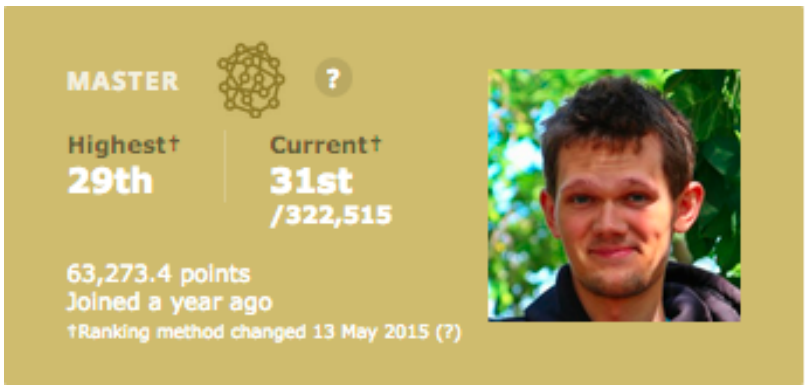


3,848 players on 3,514 teams competed to classify items across Otto Group's product lines

The Basics

What was your background prior to entering this challenge?

I have some theoretical understanding of machine learning thanks to my base institute (Moscow Institute of Physics and Technology) and our professor Konstantin Vorontsov, one of the top Russian machine learning specialists. As for my acquaintance with practical problems, another great Russian data scientist who once was Top-1 on Kaggle, Alexander D'yakonov, used to teach a course on practical machine learning every autumn which gave me very good basis. Kagglers may know this course as PZAD.



Alexander's profile on Kaggle

How did you get started competing on Kaggle?

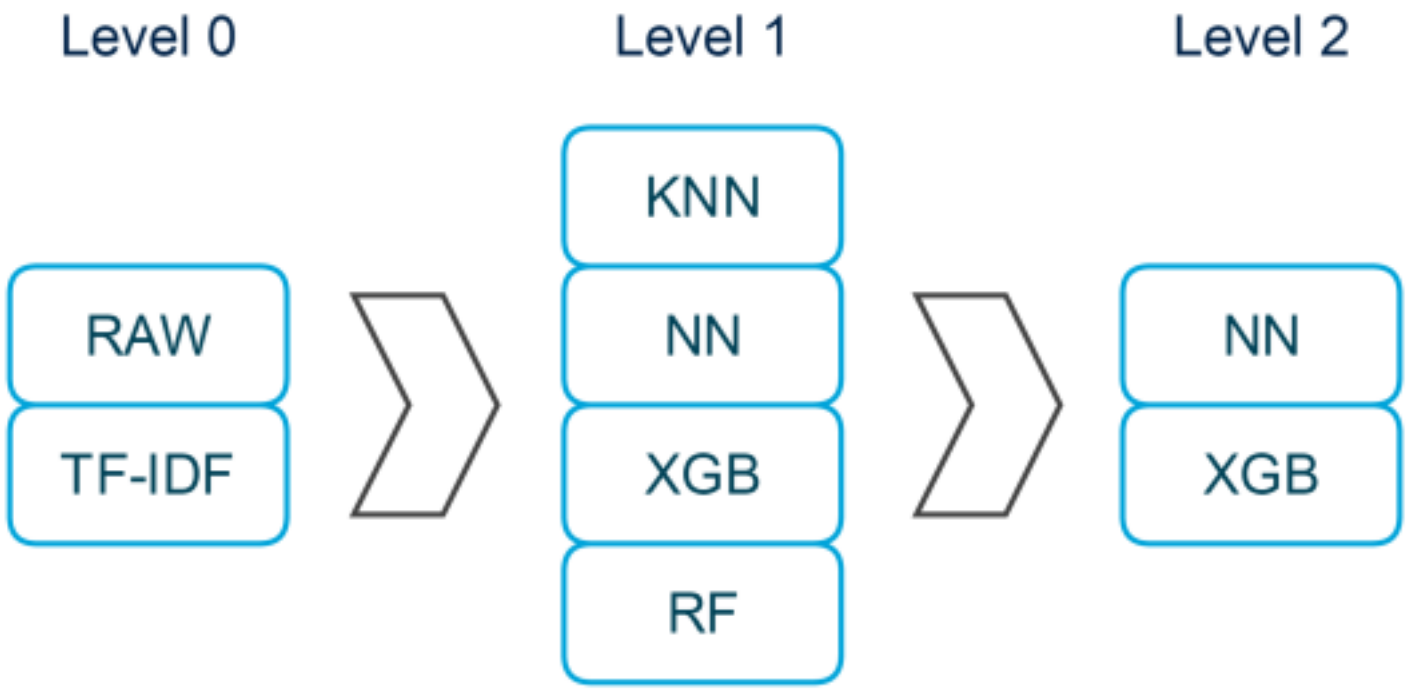
I got started in 2014's autumn in "Forest Cover Type Prediction". At that time I had no experience in solving machine learning problems. I found excellent benchmarks in "Titanic: Machine Learning from Disaster" which helped me a lot. After that I understand that machine learning is extremely interesting for me and just tried to participate in every competition I could.

What made you decide to enter this competition?

I wanted to check some ideas for my bachelor work. I liked that Otto competition has quite reliable dataset. You can check everything on cross-validation and changes on CV were close enough to leaderboard. Also, the spirit of competition is quite appropriate for checking ideas.

Let's Get Technical

What preprocessing and supervised learning methods did you use?



My solution's stacking schema

The main idea of my solution is stacking. Stacking helps you to combine different methods' predictions of Y (or labels when it comes to multiclass problems) as "metafeatures". Basically, to obtain metafeature for train, you split your data into K folds, training K models on K-1 parts while making prediction for 1 part that was left aside for each K-1 group. To obtain metafeature for test, you can average predictions from these K models or make single prediction based on all train data. After that you train metaclassifier on features & metafeatures and average predictions if you have several metaclassifiers.

In the beginning of working on the competition I found useful to split data in two groups : (1) train & test, (2) TF-IDF(train) & TF-IDF(test). Many parts of my solution use these two groups in parallel.

Talking about supervised methods, I've found that Xgboost and neural networks both give good results on data. Thus I decided to use them as metaclassifiers in my ensemble.

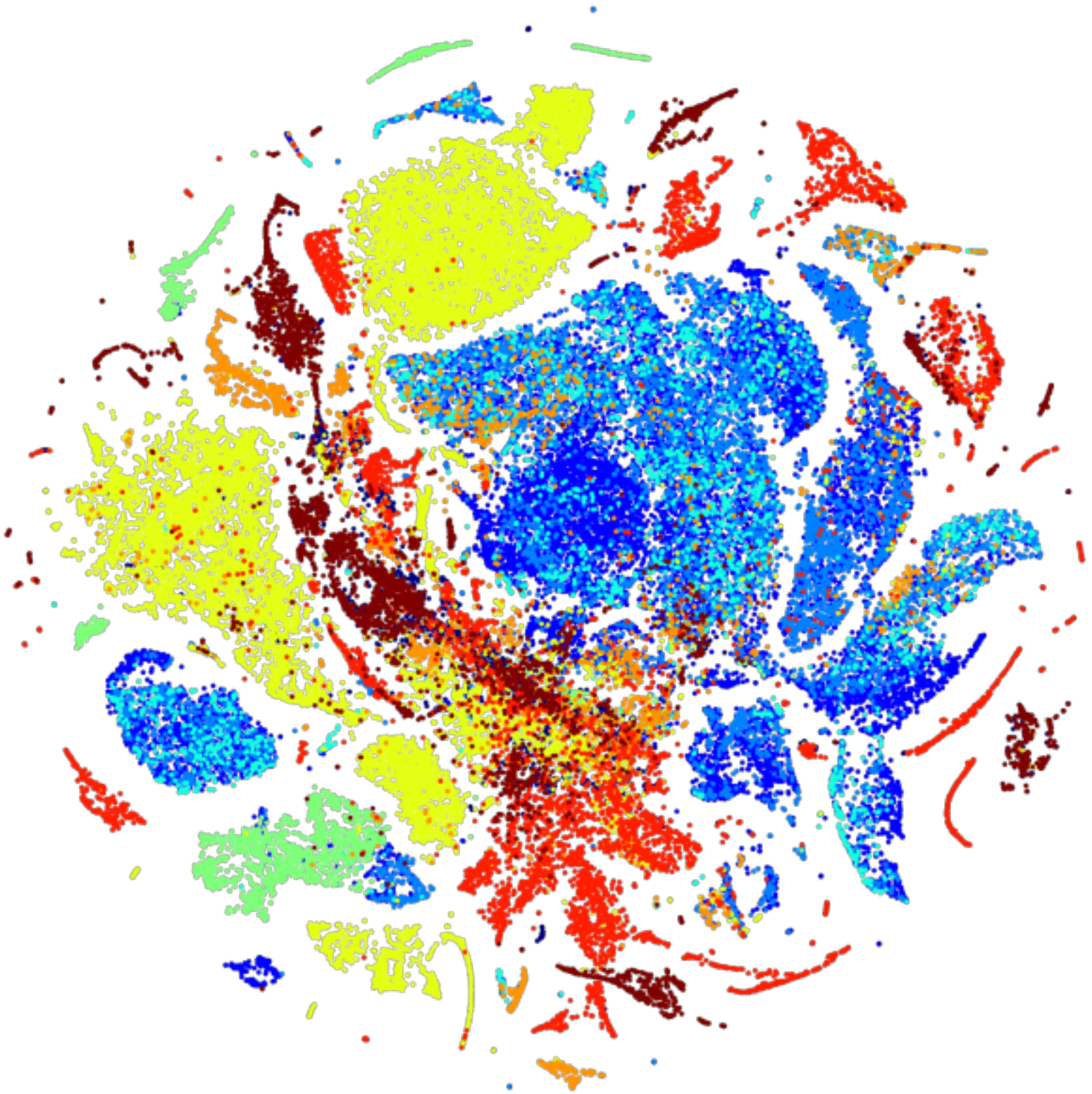
Nevertheless, KNN usually gives predictions that are very different from decision trees or neural networks, so I include them on the first level of ensemble as metafeatures. Random forest and xgboost also happened to be useful as metafeatures.

What was your most important insight into the data?

Probably the main insight was that **KNN is capable of making very good metafeatures**. Never underestimate nearest neighbours algorithm.

Very important were to combine NN and XGB predictions on the second level. While my final second- level NN and XGB separately scored around .391 on private LB, the

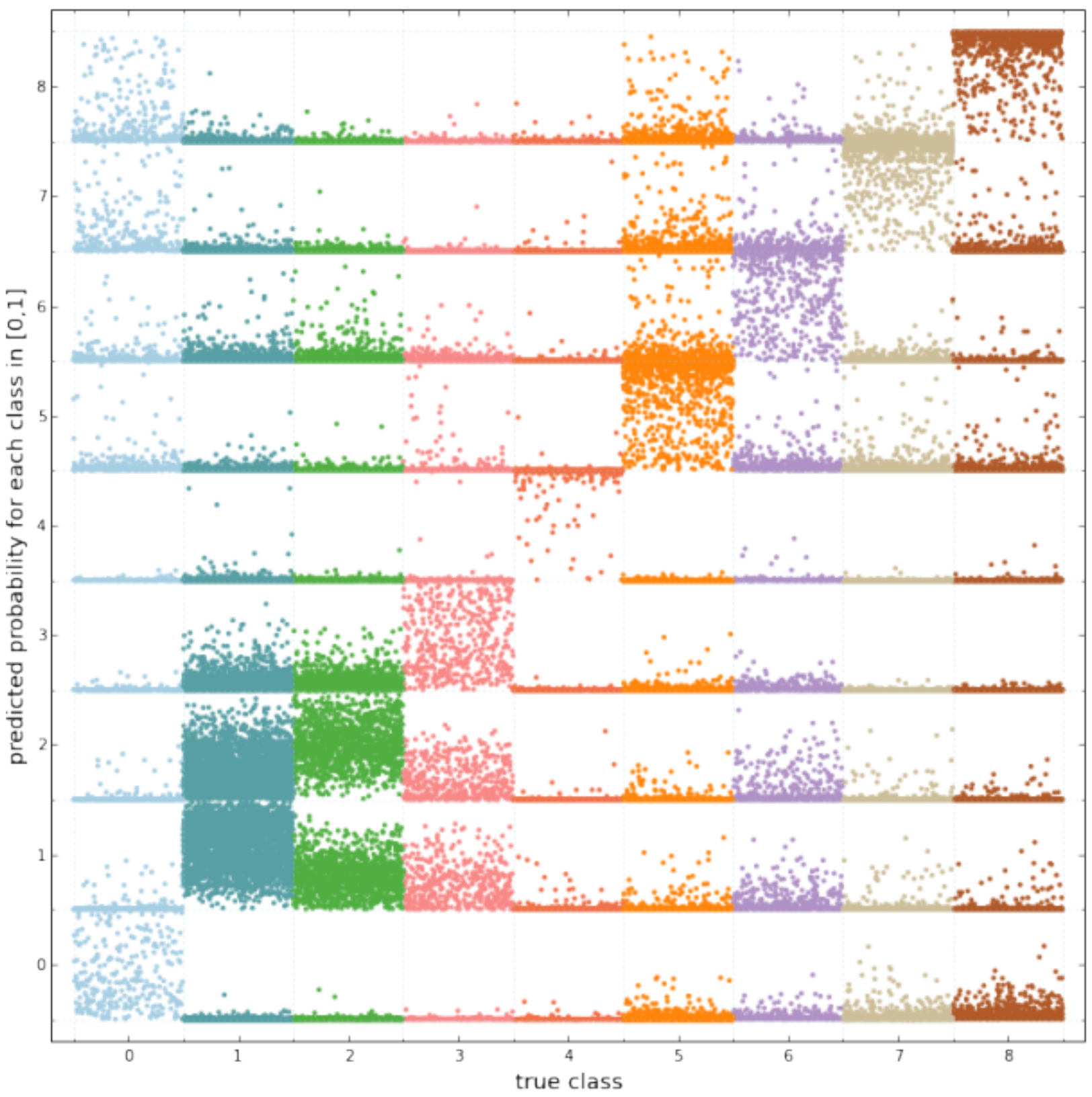
combination of them achieved .386, which is very significant improvement. Bagging on the second level helped a lot too.



TSNE in 2 dimensions

Beside this, TSNE in 2 dimensions looks very interesting. We can see on the plot that we have some examples which most likely will be misclassified by our algorithm. It does mean that it won't be easy to find a way to post-process our predictions to improve logloss.

Also, it seemed interesting that some classes were related closer than others, for example class 1 and class 2. It's worth trying to distinguish these classes specially.



Final model’s predictions for holdout’

Were you surprised by any of your findings?

Unfortunately, it appears that you won’t necessarily improve your model if you will make your metafeatures better. And when it comes to ensembling, all that you can count on is your understanding of algorithms **(basically, the more diverse metafeatures you have, the better)** and effort to try as many metafeatures as possible.

The more diverse metafeatures you have, the better. Metafeature by Extratrees vs metafeature by Neural Network.

Which tools did you use?

I only used sklearn, xgboost, lasagne. These are perfect machine learning libraries and I would recommend them to anyone who is starting to compete on Kaggle. Relying on my past experience they are sufficient to try different methods and achieve great results in most Kaggle competitions.

Words of Wisdom

Do you have any advice for those just getting started in data science?

I think that the most useful advice here is try not to stuck trying to fine-tune

parameters or stuck using the same approaches every competition. Read through forums, understand winning solutions of past competitions and all of this will give you significant boost whatever your level is. **In another words, my point is that reading past solutions is as important as solving competitions.**

Also, when you first starting to work on machine learning problems you could make some nasty mistakes which will cost you a lot of time and efforts. Thus it is great if you can work in a team with someone and ask him to check you code or try the same methods on his own. Besides always compare your performance with people on forums. **When you see that you algorithm performs much worse than people report on forum, go and check benchmarks for this and other recent competitions and try to figure out the mistake.**

Bio

Alexander Guschin

is 4th year student in Moscow Institute of Physics and Technology. Currently, Alexander is finishing his bachelor diploma work about ensembling methods.