Model Selection in Data Analysis Competitions

David Kofoed Wind[1] and **Ole Winther**[2]

Abstract. The use of data analysis competitions for selecting the most appropriate model for a problem is a recent innovation in the field of predictive machine learning. Two of the most well-known examples of this trend was the Netflix Competition and recently the competitions hosted on the online platform Kaggle.

In this paper, we will state and try to verify a set of qualitative hypotheses about predictive modelling, both in general and in the scope of data analysis competitions. To verify our hypotheses we will look at previous competitions and their outcomes, use qualitative interviews with top performers from Kaggle and use previous personal experiences from competing in Kaggle competitions.

The stated hypotheses about feature engineering, ensembling, overfitting, model complexity and evaluation metrics give indications and guidelines on how to select a proper model for performing well in a competition on Kaggle.

1 Introduction

In recent years, the amount of available data has increased exponentially and "Big Data Analysis" is expected to be at the core of most future innovations [2, 4, 5]. A new and very promising trend in the field of predictive machine learning is the use of data analysis competitions for model selection. Due to the rapid development in the field of competitive data analysis, there is still a lack of consensus and literature on how one should approach predictive modelling competitions.

In his well-known paper "Statistical Modeling : The Two Cultures" [1], Leo Breiman divides statistical modelling into two cultures, the *data modelling culture* and the *algorithmic modelling culture*. The arguments put forward in [1] justifies an approach to predictive modelling where the focus is purely on predictive accuracy. That this is the right way of looking at statistical modelling is the underlying assumption in statistical prediction competitions, and consequently also in this paper.

The concept of machine learning competitions was made popular with the Netflix Prize, a massive open competition with the aim of constructing the best algorithm for predicting user ratings of movies. The competition featured a prize of 1,000,000 dollars for the first team to improve Netflix's own results by 10% and multiple teams achieved this goal. After the success with the Netflix Prize, the website Kaggle was born, providing a platform for predictive modelling. Kaggle hosts numerous data prediction competitions and has more than 170,000 users worldwide.

The basic structure of a predictive modelling competition – as seen for example on Kaggle and in the Netflix competition – is the following: A predictive problem is described, and the participants are given a dataset with a number of samples and the true target values (the values to predict) for each sample given, this is called the training set. The participants are also given another dataset like the training set, but where the target values are not known, this is called the test set. The task of the participants is to predict the correct target values for the test set, using the training set to build their models. When participants have a set of proposed predictions for the test set, they can submit these to a website, which will then evaluate the submission on a part of the test set known as the quiz set, the validation set or simply as the public part of the test set. The result of this evaluation on the quiz set is shown in a leaderboard giving the participants an idea of how they are progressing.

Using a competitive approach to predictive modelling is being praised by some as the modern way to do science:

> Kaggle recently hosted a bioinformatics contest, which required participants to pick markers in a series of HIV genetic sequences that correlate with a change in viral load (a measure of the severity of infection). Within a week and a half, the best submission had already outdone the best methods in the scientific literature. [3]
>
> (Anthony Goldbloom, Founder and CEO at Kaggle)

> These prediction contests are changing the landscape for researchers in my area an area that focuses on making good predictions from finite (albeit sometimes large) amount of data. In my personal opinion, they are creating a new paradigm with distinctive advantages over how research is traditionally conducted in our field. [6]
>
> (Mu Zhu, Associate Professor, University of Waterloo)

This competitive approach is interesting and seems fruitful – one can even see it as an extension of the aggregation ideas put forward in [1] in the sense that the winning model is simply the model with the best accuracy, not taking computational efficiency or interpretability into account. Still one could ask if the framework provided by for example Kaggle gives a trustworthy resemblance of real-world predictive modelling problems where problems do not come with a quiz set and a leaderboard.

In this paper we state five hypotheses about building and selecting models for competitive data analysis. To verify these hypotheses we will look at previous competitions and their outcomes, use qualitative interviews with top performers from Kaggle and use previous personal experiences from competing in Kaggle competitions.

[1] Technical University of Denmark, Denmark, email: dawi@dtu.dk
[2] Technical University of Denmark, Denmark, email: olwi@dtu.dk

2 Interviews and Previous Competitions

In this section we will shortly describe the data we are using. We will list the people whom we interviewed and name the previous Kaggle competition we are using for empirical data.

2.1 Interviews

To help answer the questions we are stating, we have asked a series of questions to some of the best Kaggle participants throughout time. We have talked (by e-mail) with the following participants (name, Kaggle username, current rank on Kaggle):

- Steve Donoho (BreakfastPirate #2)
- Lucas Eustaquio (Leustagos #6)
- Josef Feigl (Josef Feigl #7)
- Zhao Xing (xing zhao #10)
- Anil Thomas (Anil Thomas #11)
- Luca Massaron (Luca Massaron #13)
- Gábor Takács (Gábor Takács #20)
- Tim Salimans (Tim Salimans #48)

Answers and parts of answers to our questions are included in this paper as quotes when relevant.

2.2 Previous competitions

Besides the qualitative interviews with Kaggle masters, we also looked at 10 previous Kaggle competitions, namely the following:

- Facebook Recruiting III - Keyword Extraction
- Partly Sunny with a Chance of Hashtags
- See Click Predict Fix
- Multi-label Bird Species Classification - NIPS 2013
- Accelerometer Biometric Competition
- AMS 2013-2014 Solar Energy Prediction Contest
- StumbleUpon Evergreen Classification Challenge
- Belkin Energy Disaggregation Competition
- The Big Data Combine Engineered by BattleFin
- Cause-effect pairs

These competitions were selected as 10 consecutive competitions, where we excluded a few competitions which did not fit the standard framework of statistical data analysis (for example challenges in optimization and operations research).

Throughout this paper, these competitions are referenced with the following abbreviated names: FACEBOOK, SUNNYHASHTAGS, SEECLICKPREDICT, BIRD, ACCELEROMETER, SOLARENERGY, STUMBLEUPON, BELKIN, BIGDATA and CAUSEEFFECT.

3 Hypotheses

In this section we state 5 hypotheses about predictive modelling in a competitive framework. We will try to verify the validity of each hypothesis using a combination of mathematical arguments, empirical evidence from previous competitions and qualitative interviews we did with some of the top participants at Kaggle. The five hypotheses to be investigated are:

1. Feature engineering is the most important part of predictive machine learning
2. Overfitting to the leaderboard is a real issue
3. Simple models can get you very far
4. Ensembling is a winning strategy
5. Predicting the right thing is important

3.1 Feature engineering is the most important part

With the extensive amount of free tools and libraries available for data analysis, everybody has the possibility of trying advanced statistical models in a competition. As a consequence of this, what gives you most "bang for the buck" is rarely the statistical method you apply, but rather the features you apply it to. By feature engineering, we mean using domain specific knowledge or automatic methods for generating, extracting, removing or altering features in the data set.

> For most Kaggle competitions the most important part is feature engineering, which is pretty easy to learn how to do. (Tim Salimans)

> The features you use influence more than everything else the result. No algorithm alone, to my knowledge, can supplement the information gain given by correct feature engineering. (Luca Massaron)

> Feature engineering is certainly one of the most important aspects in Kaggle competitions and it is the part where one should spend the most time on. There are often some hidden features in the data which can improve your performance by a lot and if you want to get a good place on the leaderboard you have to find them. If you screw up here you mostly can't win anymore; there is always one guy who finds all the secrets.
>
> However, there are also other important parts, like how you formulate the problem. Will you use a regression model or classification model or even combine both or is some kind of ranking needed. This, and feature engineering, are crucial to achieve a good result in those competitions.
>
> There are also some competitions were (manual) feature engineering is not needed anymore; like in image processing competitions. Current state of the art deep learning algorithms can do that for you. (Josef Feigl)

There are some specific types of data which have previously required a larger amount of feature engineering, namely text data and image data. In many of the previous competitions with text and image data, feature engineering was a huge part of the winning solutions (examples of this are for example SUNNYHASHTAGS, FACEBOOK, SEECLICKPREDICT and BIRD). At the same time (perhaps due to the amount of work needed to do good feature engineering here) deep learning approaches to automatic feature extraction have gained popularity.

In the competition SUNNYHASHTAGS which featured text data taken from Twitter, feature engineering was a major part of the winning solution. The winning solution used a simple regularized regression model, but generated a lot of features from the text:

> My set of features included the basic tfidf of 1,2,3-grams and 3,5,6,7 ngrams. I used a CMU Ark Twitter dedicated tokenizer which is especially robust for processing tweets + it tags the words with part-of-speech tags which can be useful to derive additional features. Additionally, my base feature set included features derived from sentiment dictionaries that map

> each word to a positive/neutral/negative sentiment. I found this helped to predict S categories by quite a bit. Finally, with Ridge model I found that doing any feature selection was only hurting the performance, so I ended up keeping all of the features ~ 1.9 mil. The training time for a single model was still reasonable. (aseveryn - 1st place winner)

In the competitions which did not have text or image data, feature engineering sometimes still played an important role in the winning entries. An example of this is the CAUSEEFFECT competition, where the winning entry created thousands of features, and then used genetic algorithms to remove non-useful features again. On the contrary, sometimes the winning solutions are those which go a non-intuitive way and simply use a black-box approach. An example of this is the SOLARENERGY competition where the Top-3 entries almost did not use any feature engineering (even though this seemed like the most intuitive approach for many) – and simply combined the entire dataset into one big table and used a complex black-box model.

Having too many features (making the feature set overcomplete), is not advisable either, since redundant or useless features tend to reduce the model accuracy.

3.1.1 Mathematical justification for feature engineering

When using simple models, it is often necessary to engineer new features to capture the right trends in the data. The most common example of this, is attempting to use a linear method to model non-linear behaviour.

To give a simple example of this, assume we want to predict the price of a house H given the dimensions (length l_H and width w_H of the floor plan) of the house. Assume also that the price $p(H)$ can be described as a linear function $p(H) = \alpha a_H + \beta$, where $a_H = l_H \cdot w_H$ is the area. By fitting a linear regression model to the original parameters l_H, w_H, we will not capture the quadratic trend in the data. If we instead construct a new feature $a_H = l_H \cdot w_H$ (the area), for each data sample (house), and fit a linear regression model using this new feature, then we will be able to capture the trend we are looking for.

3.2 Simple models can get you very far

When looking through descriptions of people's solutions after a competition has ended, there is often a surprising number of very simple solutions obtaining good results. What is also (initially) surprising, is that the simplest approaches are often described by some of the most prominent competitors.

> I think beginners sometimes just start to "throw" algorithms at a problem without first getting to know the data. I also think that beginners sometimes also go too-complex-too-soon. There is a view among some people that you are smarter if you create something really complex. I prefer to try out simpler. I "try" to follow Albert Einsteins advice when he said, "Any intelligent fool can make things bigger and more complex. It takes a touch of genius – and a lot of courage – to move in the opposite direction". (Steve Donoho)

> My first few submissions are usually just "baseline" submissions of extremely simple models – like "guess the average" or "guess the average segmented by variable X". These are simply to establish what is possible with very simple models. You'd be surprised that you can sometimes come very close to the score of someone doing something very complex by just using a simple model. (Steve Donoho)

> I think a simple model can make you top 10 in a Kaggle competition. In order to get a money prize, you have to go to ensembles most of time. (Zhao Xing)

> You can go very far [with simple models], if you use them well, but likely you cannot win a competition by a simple model alone. Simple models are easy to train and to understand and they can provide you with more insight than more complex black boxes. They are also easy to be modified and adapted to different situations. They also force you to work more on the data itself (feature engineering, data cleaning, missing data estimation). On the other hand, being simple, they suffer from high bias, so they likely cannot catch a complex mapping of your unknown function. (Luca Massaron)

Simplicity can come in multiple forms, both regarding the complexity of the model, but also regarding the pre-processing of the data. In some competitions, regularized linear regression can be the winning model in spite of its simplicity. In other cases, the winning solutions are those who do almost no pre-processing of the data (as seen in for example the SOLARENERGY competition).

3.3 Ensembling is a winning strategy

As described in [1], complex models and in particular models which are combinations of many models should perform better when measured on predictive accuracy. This hypothesis can be backed up by looking at the winning solutions for the latest competitions on Kaggle.

If one considers the 10 Kaggle competitions mentioned in Section 2.2 and look at which models the top participants used, one finds that in 8 of the 10 competitions, model combination and ensemble-models was a key part of the final submission. The only two competitions where no ensembling was used by the top participants were FACEBOOK and BELKIN, where a possible usage of model combination was non-trivial and where the data sets were of a size that favored simple models.

> No matter how faithful and well tuned your individual models are, you are likely to improve the accuracy with ensembling. Ensembling works best when the individual models are less correlated. Throwing a multitude of mediocre models into a blender can be counterproductive. Combining a few well constructed models is likely to work better. Having said that, it is also possible to overtune an individual model to the detriment of the overall result. The tricky part is finding the right balance. (Anil Thomas)

> [The fact that most winning entries use ensembling] is natural from a competitors perspective, but potentially very hurtful for Kaggle/its clients: a solution consisting of an ensemble of 1000 black box models does not give any insight and will be extremely difficult to reproduce. This will not translate to real business value for the comp organizers. (Tim Salimans)

> I am a big believer in ensembles. They do improve accuracy. BUT I usually do that as a very last step. I usually try to squeeze all that I can out of creating derived variables and using individual algorithms. After I feel like I have done all that I can on that front, I try out ensembles.
>
> (Steve Donoho)

> Ensembling is a no-brainer. You should do it in every competition since it usually improves your score. However, for me it is usually the last thing I do in a competition and I don't spend too much time on it. (Josef Feigl)

Besides the intuitive appeal of averaging models, one can justify ensembling mathematically.

3.3.1 Mathematical justification for ensembling

To justify ensembling mathematically, we refer to the approach of [7]. They look at a *one-of-K* classification problem and model the probability of input x belonging to class i as

$$f_i(x) = p(c_i|x) + \beta_i + \eta_i(x),$$

where $p(c_i|x)$ is an a posteriori probability distribution of the i-th class given input x, where β_i is a bias for the i-th class (which is independent of x) and where $\eta_i(x)$ is the error of the output for class i.

They then derive the following expression for how the added error (the part of the error due to our model fit being wrong) changes when averaging over the different models in the ensemble:

$$E_{\text{add}}^{\text{ave}} = E_{\text{add}} \left(\frac{1 + \delta(N-1)}{N} \right),$$

where δ is the average correlation between the models (weighted by the prior probabilities of the different classes) and N is the number of models trained.

The important take-away from this result is that ensembling works best if the models we combine have a low correlation. A key thing to note though, is that low correlation between models in itself is not enough to guarantee a lowering of the overall error. Ensembling as described above is effective in lowering the variance of a model but not in lowering the bias.

3.4 Overfitting to the leaderboard is an issue

During a competition on Kaggle, the participants have the possibility of submitting their solutions (predictions on the public and private test set) to a public leaderboard. By submitting a solution to the leaderboard you get back an evaluation of your model on the public-part of the test set. It is clear that obtaining evaluations from the leaderboard gives you additional information/data, but it also introduces the possibility of overfitting to the leaderboard-scores:

> The leaderboard definitely contains information. Especially when the leaderboard has data from a different time period than the training data (such as with the heritage health prize). You can use this information to do model selection and hyperparameter tuning. (Tim Salimans)

> The public leaderboard is some help, [...] but one needs to be careful to not overfit to it especially on small datasets. Some masters I have talked to pick their final submission based on a weighted average of their leaderboard score and their CV score (weighted by data size). Kaggle makes the dangers of overfit painfully real. There is nothing quite like moving from a good rank on the public leaderboard to a bad rank on the private leaderboard to teach a person to be extra, extra careful to not overfit. (Steve Donoho)

> Having a good cross validation system by and large makes it unnecessary to use feedback from the leaderboard. It also helps to avoid the trap of overfitting to the public leaderboard.
>
> (Anil Thomas)

> Overfitting to the leaderboard is always a major problem. The best way to avoid it is to completely ignore the leaderboard score and trust only your cross-validation score. The main problem here is that your cross-validation has to be correct and that there is a clear correlation between your cv-score and the leaderboard score (e.g. improvement in your cv-score lead to improvement on the leaderboard). If that's the case for a given competition, then it's easy to avoid overfitting. This works usually well if the test set is large enough.
>
> If the testset is only small in size and if there is no clear correlation, then it's very difficult to only trust your cv-score. This can be the case if the test set is taken from another distribution than the train set. (Josef Feigl)

In the 10 last competitions on Kaggle, two of them showed extreme cases of overfitting and four showed mild cases of overfitting. The two extreme cases were BIGDATA and STUMBLEUPON. In Table 1 the Top-10 submissions on the public test set from BIGDATA is shown, together with the results of the same participants on the private test set.

Name	# Public	# Private	Public score	Private score
Konstantin Sofiyuk	1	378	0.40368	0.43624
Ambakhof	2	290	0.40389	0.42748
SY	3	2	0.40820	0.42331
Giovanni	4	330	0.40861	0.42893
asdf	5	369	0.41078	0.43364
dynamic24	6	304	0.41085	0.42782
Zoey	7	205	0.41220	0.42605
GKHI	8	288	0.41225	0.42746
Jason Sumpter	9	380	0.41262	0.44014
Vikas	10	382	0.41264	0.44276

Table 1. Results of the Top-10 participants on the leaderboard for the competition: "Big Data Combine"

In BIGDATA, the task was to predict the value of stocks multiple hours into the future, which is generally thought to be extremely difficult [3]. The extreme jumps on the leaderboard is most likely due to the sheer difficulty of predicting stocks combined with overfitting.

In the cases where there were small differences between the public leaderboard and the private leaderboard, the discrepancy can also sometimes be explained by the scores for the top competitors being so close that random noise affected the positions.

[3] This is similar to what is known as the Efficient Market Hypothesis.

3.5 Predicting the right thing is important

One task that is sometimes trivial, and other times not, is that of "predicting the right thing". It seems quite trivial to state that it is important to predict the right thing, but it is not always a simple matter in practice.

> A next step is to ask, "What should I actually be predicting?". This is an important step that is often missed by many – they just throw the raw dependent variable into their favorite algorithm and hope for the best. But sometimes you want to create a derived dependent variable. I'll use the GE Flightquest as an example you dont want to predict the actual time the airplane will land; you want to predict the length of the flight; and maybe the best way to do that is to use that ratio of how long the flight actually was to how long it was originally estimate to be and then multiply that times the original estimate. (Steve Donoho)

There are two ways to address the problem of predicting the right thing: The first way is the one addressed in the quote from Steve Donoho, about predicting the correct derived variable. The other is to train the statistical models using the appropriate loss function.

> Just moving from RMSE to MAE can drastically change the coefficients of a simple model such as a linear regression. Optimizing for the correct metric can really allow you to rank higher in the LB, especially if there is variable selection involved. (Luca Massaron)

> Usually it makes sense to optimize the correct metric (especially in your cv-score). [...] However, you don't have to do that. For example one year ago, I've won the Event Recommendation Engine Challenge which metric was MAP. I never used this metric and evaluated all my models using LogLoss. It worked well there. (Josef Feigl)

As an example of why using the wrong loss function might give rise to issues, look at the following simple example: Say you want to fit the simplest possible regression model, namely just an intercept a to the data:

$$x = (0.1,\ 0.2,\ 0.4,\ 0.2,\ 0.2,\ 0.1,\ 0.3,\ 0.2,\ 0.3,\ 0.1,\ 100)$$

If we let a_{MSE} denote the a minimizing the mean squared error, and let a_{MAE} denote the a minimizing the mean absolute error, we get the following

$$a_{\text{MSE}} \approx 9.2818, \qquad a_{\text{MAE}} \approx 0.2000$$

If we now compute the MSE and MAE using both estimates of a, we get the following results:

$$\frac{1}{11}\sum_i |x_i - a_{\text{MAE}}| \approx 9 \qquad \frac{1}{11}\sum_i |x_i - a_{\text{MSE}}| \approx 16$$

$$\frac{1}{11}\sum_i (x_i - a_{\text{MAE}})^2 \approx 905 \qquad \frac{1}{11}\sum_i (x_i - a_{\text{MSE}})^2 \approx 822$$

We see (as expected) that for each loss function (MAE and MSE), the parameter which was fitted to minimize that loss function achieves a lower error. This should come as no surprise, but when the loss functions and statistical methods become complicated (such as Normalized Discounted Cumulative Gain used for some ranking competitions), it is not always as trivial to see if one is actually optimizing the correct thing.

4 Additional advice

In addition to the quotes related to the five hypotheses, the top Kaggle-participants also revealed helpful comments for performing well in a machine learning competition. Some of their statements are given in this section.

> The best tip for a newcomer is the read the forums. You can find a lot of good advices there and nowaday also some code to get you started. Also, one shouldn't spend too much time on optimizing the parameters of the model at the beginning of the competition. There is enough time for that at the end of a competition. (Josef Feigl)

> In each competiton I learn a bit more from the winners. A competiton is not won by one insight, usually it is won by several careful steps towards a good modelling approach. Everything play its role, so there is no secret formula here, just several lessons learned applied together. I think new kagglers would benefit more of carefully reading the forums and the past competitions winning posts. Kaggle masters aren't cheap on advice! (Lucas Eustaquio)

> My most surprising experience was to see the consistently good results of Friedman's gradient boosting machine. It does not turn out from the literature that this method shines in practice. (Gabor Takacs)

> The more tools you have in your toolbox, the better prepared you are to solve a problem. If I only have a hammer in my toolbox, and you have a toolbox full of tools, you are probably going to build a better house than I am. Having said that, some people have a lot of tools in their toolbox, but they don't know *when* to use *which* tool. I think knowing when to use which tool is very important. Some people get a bunch of tools in their toolbox, but then they just start randomly throwing a bunch of tools at their problem without asking, "Which tool is best suited for this problem?" (Steve Donoho)

5 Conclusion

This paper looks at the recent trend of using data analysis competitions for selecting the most appropriate model for a specific problem. When participating in data analysis competitions, models get evaluated solely based on their predictive accuracy. Because the submitted models are not evaluated on their computational efficiency, novelty or interpretability, the model construction differs slightly from the way models are normally constructed for academic purposes and in industry.

We stated a set of five different hypotheses about the way to select and construct models for competitive purposes. We then used a combination of mathematical theory, experience from past competitions and qualitative interviews with top participants from Kaggle to try and verify these hypotheses.

Although there is no secret formula for winning a data analysis competition, the stated hypotheses together with additional good advice from top performing Kaggle competitors, give indications and guidelines on how to select a proper model for performing well in a competition on Kaggle.

REFERENCES

[1] Leo Breiman, 'Statistical modeling: The two cultures', *Statistical Science*, (2001).

[2] World Economic Forum. Big data, big impact: New possibilities for international development. `http://bit.ly/1fbP4aj`, January 2012. [Online].

[3] A. Goldbloom, 'Data prediction competitions – far more than just a bit of fun', in *Data Mining Workshops (ICDMW), 2010 IEEE International Conference on*, pp. 1385–1386, (Dec 2010).

[4] Steve Lohr. The age of big data. `http://www.nytimes.com/2012/02/12/sunday-review/big-datas-impact-in-the-world.html`, February 2012. [Online; posted 11-February-2012].

[5] James Manyika, Michael Chui, Brad Brown, Jacques Bughin, Richard Dobbs, Charles Roxburgh, and Angela Hung Byers. Big data: The next frontier for innovation, competition and productivity. `http://www.mckinsey.com/insights/business_technology/big_data_the_next_frontier_for_innovation`, May 2011. [Online; posted May-2011].

[6] Zhum Mu. The impact of prediction contests, 2011.

[7] K. Tumer and J. Ghosh, 'Error correlation and error reduction in ensemble classifiers', *Connection Science*, **8**(3-4), 385–403, (1996).