Introduction

Machine Learning is tricky. No matter how many books you read, tutorials you finish or problems you solve, there will always be a data set you might come across where you get clueless. Specially, when you are in your early days of Machine Learning. Isn't it ?

In this blog post, you'll learn some essential tips on building machine learning models which most people learn with experience. These tips were shared by Marios Michailidis (a.k.a Kazanova), Kaggle Grandmaster, Current Rank #3 in a webinar happened on 5th March 2016. The webinar had three aspects:

1. Video - Watch Here.
2. Slides - Slides used in the video were shared by Marios. Indeed, an enriching compilation of machine learning knowledge. Below are the slides.
3. Q & As - This blog enlists all the questions asked by participants at webinar.

Questions & Answers

1. What are the steps you follow for solving a ML problem? Please describe from scratch.

Following are the steps I undertake while solving any ML problem:

1. **Understand the data** - After you download the data, start exploring features. Look at data types. Check variable classes. Create some univariate - bivariate plots to understand the nature of variables.
2. **Understand the metric to optimise** - Every problem comes with a unique evaluation metric. It's imperative for you to understand it, specially how does it change with target variable.
3. **Decide cross validation strategy** - To avoid overfitting, make sure you've set up a cross validation strategy in early stages. A nice CV strategy will help you get reliable score on leaderboard.
4. **Start hyper parameter tuning** - Once CV is at place, try improving model's accuracy using hyper parameter tuning. It further includes the following steps:
    - **Data transformations**: It involve steps like scaling, removing outliers, treating null values, transform categorical variables, do feature selections, create interactions etc.
    - **Choosing algorithms** and tuning their hyper parameters: Try multiple algorithms to understand how model performance changes.
    - **Saving results**: From all the models trained above, make sure you save their predictions. They will be useful for ensembling.
    - **Combining models**: At last, ensemble the models, possibly on multiple levels. Make sure the models are correlated for best results.

2. What are the model selection and data manipulation techniques you follow to solve a problem?

Generally, I try (almost) everything for most problems. In principle for:

1. **Time series**: I use GARCH, ARCH, regression, ARIMA models etc.
2. **Image classification**: I use deep learning (convolutional nets) in python.
3. **Sound Classification**: Common neural networks
4. **High cardinality categorical (like text data)**: I use linear models, FTRL, Vowpal wabbit, LibFFM, libFM, SVD etc.
5. For everything else,I use Gradient boosting machines (like XGBoost and LightGBM) and deep learning (like keras, Lasagne, caffe, Cxxnet). I decide what model to keep/drop in Meta modelling with feature selection techniques. Some of the feature selection techniques I use includes:
    - Forward (cv or not) - Start from null model. Add one feature at a time and check CV accuracy. If it improves keep the variable, else discard.
    - Backward (cv or not) - Start from full model and remove variables one by one. It CV accuracy improves by removing any variable, discard it.
    - Mixed (or stepwise) - Use a mix of above to techniques.
    - Permutations
    - Using feature importance - Use random forest, gbm, xgboost feature selection feature.
    - Apply some stats' logic such as chi-square test, anova.

Data manipulation technique could be different for every problem :

- **Time series** : You can calculate moving averages, derivatives. Remove outliers.
- **Text** : Useful techniques are tfidf, countvectorizers, word2vec, svd (dimensionality reduction). Stemming, spell checking, sparse matrices, likelihood encoding, one hot encoding (or dummies), hashing.
- **Image classification**: Here you can do scaling, resizing, removing noise (smoothening), annotating etc
- **Sounds** : Calculate Furrier Transforms , MFCC (Mel frequency cepstral coefficients), Low pass filters etc
- **Everything else** : Univariate feature transformations (like log +1 for numerical data), feature selections, treating null values, removing outliers, converting categorical variables to numeric.

3. Can you elaborate cross validation strategy?

Cross validation means that from my main set, I create RANDOMLY 2 sets. I built (train) my algorithm with the first one (let's call it training set) and score the other (let's call it validation set). I repeat this process multiple times and always check how my model performs on the test set in respect to the metric I want to optimise.

The process may look like:

- For 10 (you choose how many X) times
- Split the set in training (50%-90% of the original data)
- And validation (50%-10% of the original data)
- Then fit the algorithm on the training set
- Score the validation set.
- Save the result of that scoring in respect to the chosen metric.
- Calculate the average of these 10 (X) times. That how much you expect this score in real life and is generally a good estimate.
- Remember to use a SEED to be able to replicate these X splits
- Other things to consider is Kfold and stratified KFold . Read here. For time sensitive data, make certain you always the rule of having past predicting future when testing's.

4. Can you please explain some techniques used for cross validation?

- Kfold
- Stratified Kfold
- Random X% split
- Time based split
- For large data, just one validation set could suffice (like 20% of the data – you don't need to do multiple times).

5. How did you improve your skills in machine learning? What training strategy did you use?

I did a mix of stuff in 2. Plus a lot of self-research. Alongside, programming and software (in java) and A LOT of Kaggling ☺

6. Which are the most useful python libraries for a data scientist ?

Below are some libraries which I find most useful in solving problems:

- Data Manipulation
- Numpy
- Scipy
- Pandas
- Data Visualization
- Matplotlib
- Machine Learning / Deep Learning
- Xgboost
- Keras

- Nolearn
- Gensim
- Scikit image
- Natural Language Processing
- NLTK

7. What are useful ML techniques / strategies to impute missing values or predict categorical label when all the variables are categorical in nature.

Imputing missing values is a critical step. Sometimes you may find a trend in missing values. Below are some techniques I use:

- Use mean, mode, median for imputation
- Use a value outside the range of the normal values for a variable. like -1 ,or -9999 etc.
- Replace with a likelihood – e.g. something that relates to the target variable.
- Replace with something which makes sense. For example: sometimes null may mean zero
- Try to predict missing values based on subsets of know values
- You may consider removing rows with many null values

8. Can you elaborate what kind of hardware investment you have done i.e. your own PC/GPU setup for Deep learning related tasks? Or were you using more cloud based GPU services?

I won my first competition (Acquired valued shoppers challenge) and entered kaggle's top 20 after a year of continued participation on 4 GB RAM laptop (i3). I was using mostly self-made solutions up to this point (in Java). That competition it had something like 300,000,000 rows of data of transactions you had to aggregate so I had to parse the data and be smart to keep memory usage at a minimum.

However since then I made some good investments to become Rank #1. Now, I have access to linux servers of 32 cores and 256 GBM of RAM. I also have a geforce 670 machine (for deep learning /gpu tasks) . Also, I use mostly Python now. You can consider Amazon's AWS too, however this is mostly if you are really interested in getting to the top, because the cost may be high if you use it a lot.

9. Do you use high performing machine like GPU. or for example do you do thing like grid search for parameters for random forest(say), which takes lot of time, so which machine do you use?

I use GPUs (not very fast, like a geforce 670) for every deep learning training model. I have to state that for deep learning GPU is a MUST. Training neural nets on CPUs takes ages, while a mediocre GPU can make a simple nn (e.g deep learning) 50-70 times faster. I don't like grid search. I do this fairly manually. I think in the beginning it might be slow, but after a while you can get to decent solutions with the first set of parameters! That is because you can sort of learn which parameters are best for each problem and you get to know the algorithms better this way.

10. How do people built around 80+ models is it by changing the hyper parameter tuning ?

It takes time. Some people do it differently. I have some sets of params that worked in the past and I initialize with these values and then I start adjusting them based on the problem at hand. Obviously you need to forcefully explore more areas (of hyper params in order to know how they work) and enrich this bank of past successful hyper parameter combinations for each model. You should consider what others are doing too. There is NO only 1 optimal set of hyper params. It is possible you get a similar score with a completely different set of params than the one you have.

11. How does one improve their kaggle rank? Sometimes I feel hopeless while working on any competition.

It's not an overnight process. Improvement on kaggle or anywhere happens with time. There are no shortcuts. You need to just keep doing things. Below are some of the my recommendations:

- Learn better programming: Learn python if you know R.
- Keep learning tools (listed below)
- Read some books.

- Play in ‘knowledge’ competitions
- See what the others are doing in kernels or in past competitions look for the ‘winning solution sections’
- Team up with more experience users, but you need to improve your ranking slightly before this happens
- Create a code bank
- Play … a lot!

12. Can you tell us about some useful tools used in machine learning ?

Below is the list of my favourite tools:

- Liblinear : For linear models
- LibSvm for Support Vector machines
- Scikit Learn for all machine learning models
- Xgboost for fast scalable gradient boosting
- LightGBM
- Vowpal Wabbit for fast memory efficient linear models
- H2O in R for many models
- LibFm
- LibFFM
- Weka in Java (has everything)
- Graphchi for factorizations
- GraphLab for lots of stuff
- Cxxnet : One of the best implementation of convolutional neural nets out there. Difficult to install and requires GPU with NVDIA Graphics card.
- RankLib: The best library out there made in java suited for ranking algorithms (e.g. rank products for customers) that supports optimization fucntions like NDCG.
- Keras and Lasagne for neural nets. This assumes you have Theano or Tensorflow.

13. How to start with machine learning?

I like these slides from the university of utah in terms of understanding some basic algorithms and concepts about machine learning. This book for python. I like this book too. Don't forget to follow the wonderful scikit learn documentation. Use jupyter notebook from anaconda.

You can find many good links that have helped me in kaggle here. Look at 'How Did you Get Better at Kaggle'

In addition, you should do Andrew Ng's machine learning course. Alongside, you can follow some good blogs such as mlwave, fastml, analyticsvidhya. But the best way is to get your hands dirty. do some kaggle! tackle competitions that have the "knowledge" flag first and then start tackling some of the main ones. Try to tackle some older ones too.

14. What techniques perform best on large data sets on Kaggle and in general ? How to tackle memory issues ?

Big data sets with high cardinality can be tackled well with linear models. Consider sparse models. Tools like vowpal wabbit. FTRL , libfm, libffm, liblinear are good tools matrices in python (things like csr matrices). Consider ensembling (like combining) models trained on smaller parts of the data.

15. What is the SDLC (Sofware Development Life Cycle) of projects involving Machine Learning ?

- Give a walk-through on an industrial project and steps involved, so that we can get an idea how they are used. Basically, I am in learning phase and would expect to get an industry level exposure.
- Business questions: How to recommend products online to increase purchases.
- Translate this into an ml problem. Try to predict what the customer will buy in the future given some data available at the time the customer is likely to make the click/purchase, given some historical exposures to recommendations
- Establish a test /validation framework.
- Find best solutions to predict best what customer chose.

- Consider time/cost efficiency as well as performance
- Export model parameters/pipeline settings
- Apply these in an online environment. Expose some customers but NOT all. Keep test and control groups
- Assess how well the algorithm is doing and make adjustments over time.

16. Which is your favorite machine learning algorithm?

It has to be Gradient Boosted Trees. All may be good though in different tasks.

15. Which language is best for deep learning, R or Python?

I prefer Python. I think it is more program-ish . R is good too.

16. What would someone trying to switch careers in data science need to gain aside from technical skills? As I don't have a developer background would personal projects be the best way to showcase my knowledge?

The ability to translate business problems to machine learning, and transforming them into solvable problems.

17. Do you agree with the statement that in general feature engineering (so exploring and recombining predictors) is more efficient than improving predictive models to increase accuracy?

In principle – Yes. I think model diversity is better than having a few really strong models. But it depends on the problem.

18. Are the skills required to get to the leaderboard top on Kaggle also those you need for your day-to day job as a data scientist? Or do they intersect or are somewhat different? Can I make the idea of what a data scientist's job is based on Kaggle competitions? And if a person does well on Kaggle does it follow that she will be a successful data scientist in her career ?

There is some percentage of overlap especially when it comes to making predictive models, working with data through python/R and creating reports and visualizations. What Kaggle does not offer (but you can get some idea) is:

- How to translate a business question to a modelling (possibly supervised) problem
- How to monitor models past their deployment
- How to explain (many times) difficult concepts to stake holders.
- I think there is always room for a good kaggler in the industry world. It is just that data science can have many possible routes. It may be for example that not everyone tends to be entrepreneurial in their work or gets to be very client facing, but rather solving very particular (technical) tasks.

19. Which machine learning concepts are must to have to perform well in a kaggle competition?

- Data interrogation/exploration
- Data transformation – pre-processing
- Hands on knowledge of tools
- Familiarity with metrics and optimization
- Cross Validation
- Model Tuning
- Ensembling

20. How do you see the future of data scientist job? Is automation going to kill this job?

No – I don’t think so. This is what they used to say about automation through computing. But ended up requiring a lot of developers to get the job done! It may be possible that data scientists focus on softer tasks over time like translating business questions to ml problems and generally becoming shepherds’ of the process – as in managers/supervisors of the modelling process.

21. How to use ensemble modelling in R and Python to increase the accuracy of prediction. Please quote some real life examples?

You can see my github script as I explain different Machine leaning methods based on a Kaggle competition. Also, check this ensembling guide.

22. What is best python deep learning libraries or framework for text analysis?

I like Keras (because now supports sparse data), Gensim (for word 2 vec).

23. How valuable is the knowledge gained through these competitions in real life? Most often I see competitions won by ensembling many #s of models … is this the case in real life production systems? Or are interpretable models more valuable than these monster ensembles in real productions systems?

In some cases yes – being interpretable or fast (or memory efficient) is more important. But this is likely to change over time as people will be less afraid of black box solutions and focus on accuracy.

24. Should I worry about learning about the internals about the machine learning algorithms or just go ahead and try to form an understanding of the algorithms and use them (in competitions and to solve real life business problems) ?

You don't need the internals. I don't know all the internals. It is good if you do, but you don't need to. Also there are new stuff coming out every day – sometimes is tough to keep track of it. That is why you should focus on the decent usage of any algorithm rather than over investing in one.

25. Which are the best machine learning techniques for imbalanced data?

I don't do a special treatment here. I know people find that strange. This comes down to optimizing the right metric (for me). It is tough to explain in a few lines. There are many techniques for sampling, but I never had to use. Some people are using Smote. I don't see value in trying to change the principal distribution of your target variable. You just end up with augmented or altered principal odds. If you really want a cut-off to decide on whether you should act or not – you may set it based on the principal odds.

I may not be the best person to answer this. I personally have never found it (significantly) useful to change the distribution of the target variable or the perception of the odds in the target variable. It may just be that other algorithms are better than others when dealing with this task (for example tree-based ones should be able to handle this).

26. Typically, marketing research problems have been mostly handled using standard regression techniques - linear and logistic regression, clustering, factor analyses, etc…My question is how useful are machine learning and deep learning techniques/algorithms useful to marketing research or business problems? For example how useful is say interpreting the output of a neural network to clients? Are there any resources you can refer to?

They are useful in the sense that you can most probably improve accuracy (in predicting let's say marketing response) versus linear models (like regressions). Interpreting the output is hard and in my opinion it should not be necessary as we are generally moving towards more black box and complicated solutions.

As a data scientist you should put effort in making certain that you have a way to test how good your results are on some unobserved (test) data rather trying to understand why you get the type of predictions you are getting. I do think that decompressing information from complicating models is a nice topic (and valid for research), but I don't see it as necessary.

On the other hand, companies, people, data scientists, statisticians and generally anybody who could be classified as a 'data science player' needs to get educated to accept black box solutions as perfectly normal. This may take a while, so it may be good to run some regressions along with any other modelling you are doing and generally try to provide explanatory graphs and summarized information to make a case for why your models perform as such.

27. How to build teams for collaboration on Kaggle ?

You can ask in forums (i.e in kaggle) . This may take a few competitions though before 'people can trust you'. Reason being, they are afraid of duplicate accounts (which violate competition rules), so people would prefer somebody who is proven to play fair. Assuming some time has passed, you just need to think of people you would like play with, people you think you can learn from and generally people who are likely to take different approaches than you so you can leverage the benefits of diversity when combining methods.

28. I have gone through basic machine learning course(theoretical) . Now I am starting up my practical journey , you just recommended to go through sci-kit learn docs & now people are saying TENSORFLOW is the next scikit learn , so should I go through scikit or TF is a good choice ?

I don't agree with this statement 'people are saying TENSORFLOW is the next scikit learn'. Tensorflow is a framework to do well certain machine learning tasks (like for deep learning). I think you can learn both, but I would start with scikit. I personally don't know TensorFlow , but I use tools that are based on tensor flow (for example Keras). I am lazy I guess!

29. The main challenge that I face in any competition is cleaning the data and making it usable for prediction models. How do you overcome it ?

Yeah. I join the club! After a while you will create pipelines that could handle this relatively quicker. However…you always need to spend time here.

30. How to compute big data without having powerful machine?

You should consider tools like vowpal wabbit and online solutions, where you parse everything line by line. You need to invest more in programming though.

31. What is Feature Engineering?

In short, feature engineering can be understood as:

- Feature transformation (e.g. converting numerical or categorical variables to other types)
- Feature selections
- Exploiting feature interactions (like should I combine variable A with variable B?)
- Treating null values
- Treating outliers

32. Which maths skills are important in machine learning?

Some basic probabilities along with linear algebra (e.g. vectors). Then some stats help too. Like averages, frequency, standard deviation etc.

33. Can you share your previous solutions?

See some with code and some without (just general approach).

https://www.kaggle.com/c/malware-classification/discussion/13863
http://blog.kaggle.com/2015/05/11/microsoft-malware-winners-interview-2nd-place-gert-marios-aka-kazanova/
https://github.com/kaz-Anova/ensemble_amazon
http://blog.kaggle.com/2015/12/03/dato-winners-interview-1st-place-mad-professors/
http://blog.kaggle.com/2016/04/08/homesite-quote-conversion-winners-write-up-1st-place-kazanova-faron-clobber/
https://mlwave.com/how-we-won-3rd-prize-in-crowdanalytix-copd-competition/
http://blog.kaggle.com/2016/08/31/avito-duplicate-ads-detection-winners-interview-2nd-place-team-the-quants-mikel-peter-marios-sonny/
http://blog.kaggle.com/2016/12/15/bosch-production-line-performance-competition-winners-interview-3rd-place-team-data-property-avengers-darragh-marios-mathias-stanislav/

34. How long should it take for you to build your first machine learning predictor ?

Depends on the problem (size, complexity, number of features). You should not worry about the time. Generally in the beginning you might spend much time on things that could be considered much easier later on. You should not worry about the time as it may be different for each person, given the programming, background or other experience.

35. Are there any knowledge competitions that you can recommend where you are not necessarily competing on the level as Kaggle but building your skills?

From here, both titanic and digit recognizer are good competitions to start. Titanic is better because it assumes a flat file. Digit recognizer is for image classification so it might be more advanced.

36. What is your opinion about using Weka and/or R vs Python for learning machine learning?

I like Weka. It has a good documentation– especially if you want to learn the algorithms. However I have to admit that it is not as efficient as some of the R and Python implementations. It has good coverage though. Weka has some good visualizations too – especially for some tree-based algorithms. I would probably suggest you to focus on R and Python at first unless your background is strictly in Java.

Summary

In short, succeeding in machine learning competition is all about learning new things, spending a lot of time training, feature engineering and validating models. Alongside, interact with community on forums, read blogs and learn from approach of fellow competitors.