Are not all curves lines?

Lefty G Balogh

Abstract

Based on my mid-term mini project, I wanted to have a second pass at the movie-ratings database. My research question was inspired by Ilkay Altintas. She mentioned in one of the videos, that we may need to play around a little with the parameters to improve the accuracy of the naïve Bayes prediction model.

I wanted to see if there was a way to guarantee that I could pick the best tradeoff between gains and effort. In the end, I opted for one of my old-time favourites, a Monte Carlo generator, to find an optimal solution to sample sizing vs prediction accuracy.

Motivation

Data analysis is computation heavy. Despite the fact that I have a high-end laptop, it is obvious that both memory and computational capacities place a limit on what is possible, never mind feasible, or optimal. I face similar issues in my daily job where our customers generate several petabytes, one thousand million million, bytes of data monthly. Nevertheless, they expect fast, reliable predictions based on that data.

Hence comes the question: how can I make a principled decision with regards to the size of the training set for a predictive model? Is it a linear relationship? The more I give for training the better it gets? Or is it non-linear? Can I find a cutoff points where I use less data, get the best result and save time on processing?

Dataset

I used the movie ratings database we examined in week 4 for several reasons. My mini project left me wanting to see if a machine learning model could bring out a little more insight than my manual analysis. And I was also aware that its sheer size could not be dominated. My laptop could not handle the whole dataset. I reached a computational threshold that I am not used to.

Therefore, I wanted to see, if there was a way to utilize my current results and still deliver a machine learning model that is optimal in a sense that it is reliable, while at the same time does not need to process the entire gigantic dataset for its training.

Data Preparation and Cleaning

This was one of the most challenging tasks. First I had to slice the ratings data into about one third its original size because my computer kept crashing with out of memory errors.

Second, I had to establish my own criterion of what makes an excellent film. My definition for this purpose was that the sum total of 4+ star reviews minus the sum total of the <2 star reviews had to equal 1000 or more. In essence, the absolute positive reviews had to far outweigh the negative ones.

Third, based on this I had to classify the movies into whether or not they were excellent and only then could run the learning model and the analysis..

Research Question(s)

Given my computational limitations: the dataset is memory-bound: I cannot analyse the entire ratings table, and the analysis is computationally extensive: the computation of the score values and the training of the data both take considerable amounts of time

How could I deliver a good enough model that still delivers in reasonable time?

Could I find a way to limit the size of my training data in a principled fashion?

Could I argue my choice clearly and convincingly for a real-life customer?

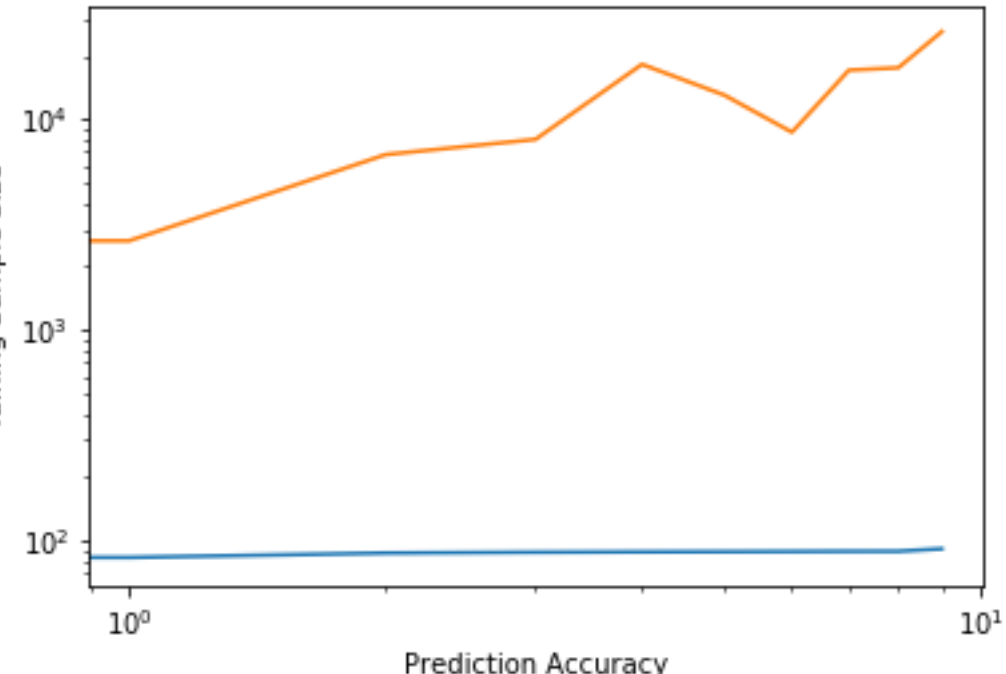
Methods

I opted for using a Monte Carlo engine to generate random runs of my learning model and then plot the randomly assigned training dataset sizes against their accuracy. This allowed me to freely experiment with minimum and maximum set sizes, the number of test runs, and with the degrees of possible as well as acceptable accuracies I could reasonable expect from my model.

I also tried several visualization methods, and eventually opted for a scatterplot as that proved to be what actually clarified to me the most the nature of the actual phenomenon, namely that accuracy in not at all a linear relationship in this game in Monte Carlo.

Findings - My First Attempt

Visualizing the relationship between training set size and accuracy



What am I seeing?

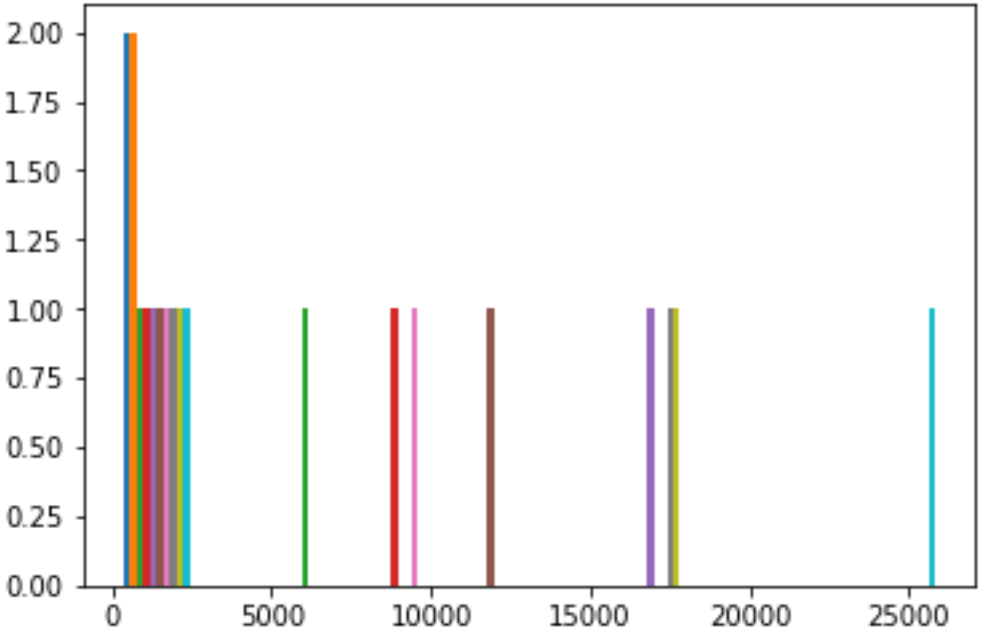
I can't show this to a customer.

I need a histogram. People understand bar charts.

That'll fix this mess.

Findings - My Second Attempt

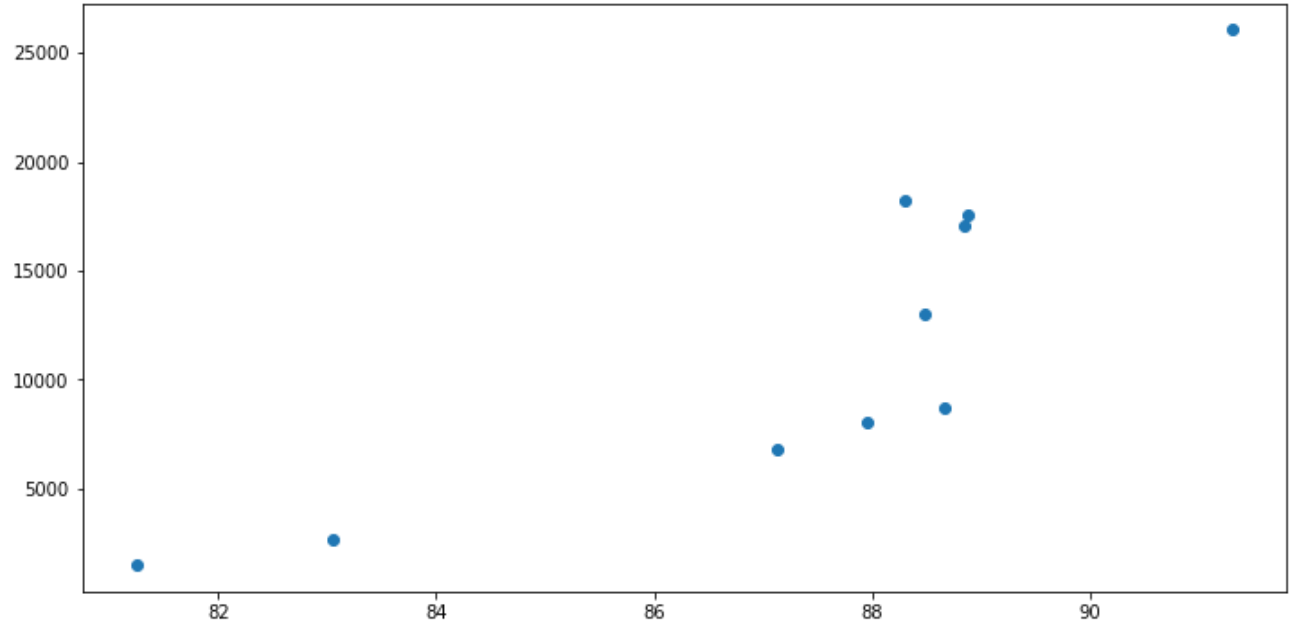
Visualizing the relationship between training set size and accuracy



Oh my… What on earth is this?
I definitely can't show THIS to a customer.
Why am I not seeing what I want to see?
What do I actually want to show people?
I want them to see the relationship between training set size and accuracy…

Findings - My Third Attempt

Visualizing the relationship between training set size and accuracy

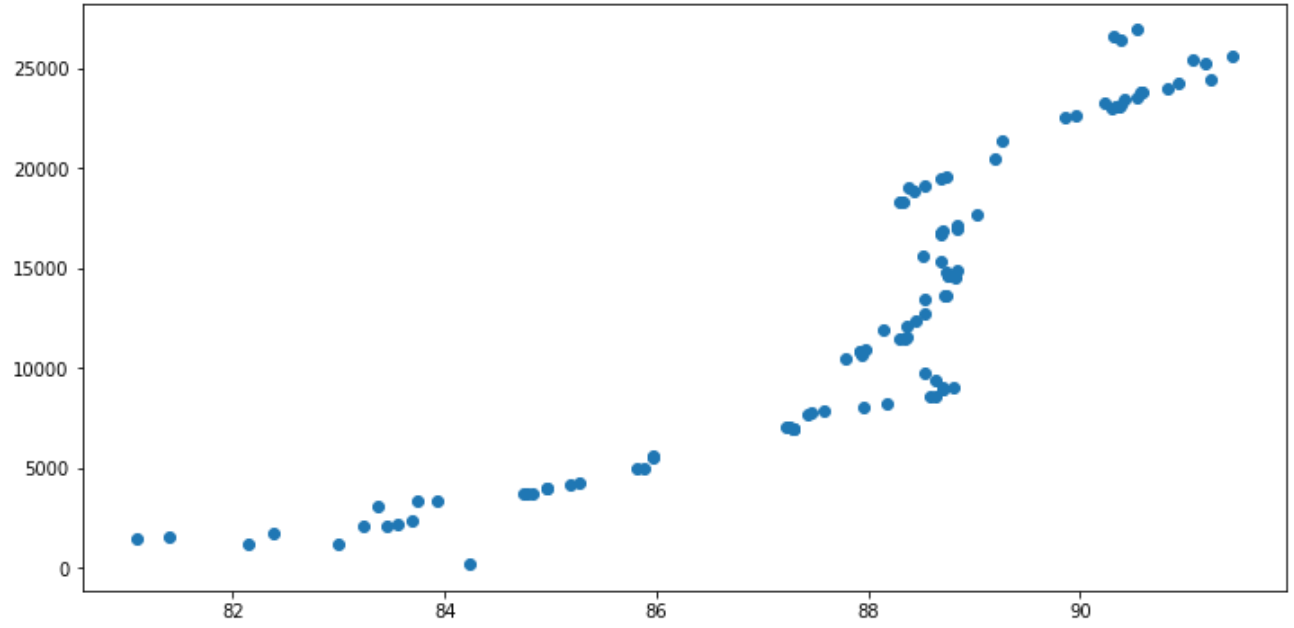


Hmmm… Ten iterations
This looks… I mean, small set, small accuracy… Big set, big accuracy…
Not linear…
Kinda clearer now.

Let’s tweak it!

Findings – 100 Iterations with Monte Carlo

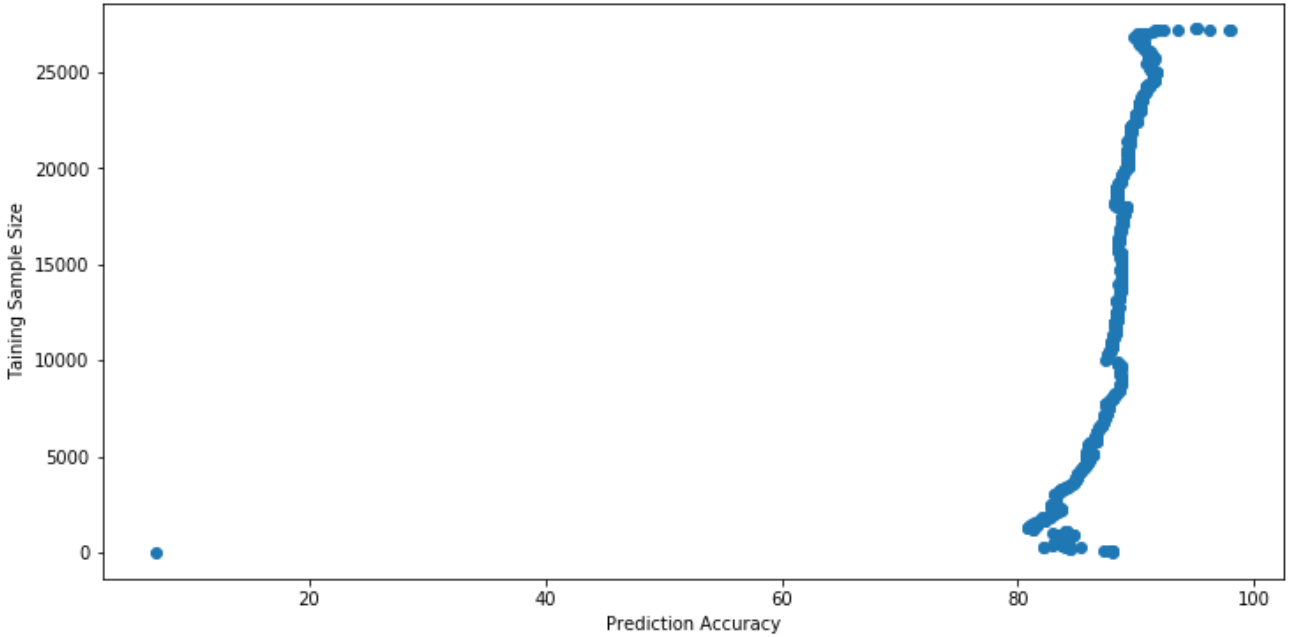
Visualizing the relationship between training set size and accuracy



Okay, so it looks that at around 10K, I begin to gain 88% accuracy. And that does not significantly improve any further.
I could add maybe another 4-5% by doubling the training set size

Findings – The Hockey Stick

Visualizing the relationship between training set size and accuracy

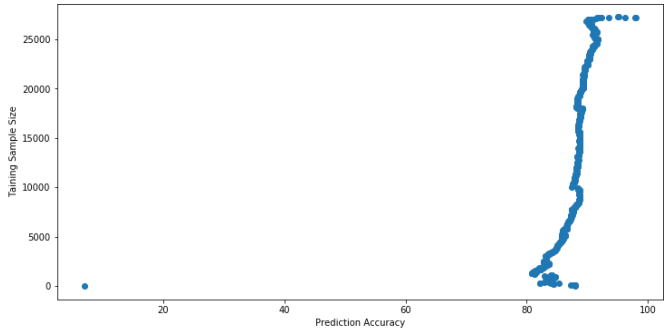


The gains with training sets over 10K would be minimal. We could even optimize at 5-6K for a low-budget solution, but a 10K set delivers an 88% accuracy.

Findings – The Hockey Stick

Summary:

- We need a few thousand entries for a fairly accurate model.
- From 4-5K to 10K, the gains are steady, so we can argue in favour of quality over cost
- From 10K to about 25K, we gain very little
- With samples almost as large as the actual entire dataset, we probably risk overfitting
- I'd recommend we settle at 10K
- Good tradeoff between accuracy and size



Limitations

The basic premise of the entire research may be faulty: do titles really predict success? Surely it must be the actual movie that delivered a 5* performance. I would need to seriously reconsider whether it really is predictive. (Good titles, of course, set expectations for genre, tone and other aspects and if the movie delivers to meet those expectations… but still…)
Furthermore I have not run the same over another slice of the ratings data set.
Finally, I have not taken genre into account for the prediction model. There maybe more in there than meet the eye in terms of predictive powers.

Conclusions

How could I deliver a good enough model that still delivers in reasonable time?

- Evaluate the gains you make in accuracy over the computational costs you incur with increasing the dataset. Do not expect a linear relationship.

Could I find a way to limit the size of my training data in a principled fashion?

- Yes. A large enough random sample can reveal an inflection point beyond which the gains are diminishing returns on investment

Conclusions

Could I argue my choice clearly and convincingly for a real-life customer?

- A good visualization empowers customers to make the right choice, so you can all sit around the table to negotiate what cost and benefits each choice entails without them having to understands that pandas are not bears

Acknowledgements

My wife, as always, deserves the biggest credit for her patience.

References

I first got introduced to the Monte Carlo engine concept in The Black Swan: The Impact of the Highly Improbable by NASSIM TALEB

The title was inspired by How Not to Be Wrong: The Power of Mathematical Thinking by Jordan Ellenberg

Both books are well worth a read.