Analysis of rank-score data for the TU Delft Aerospace Selection Process*

erentar2002

2023-04-15

The exam scores on the TU Delft Aerospace Selection process are released a day before the ranks are. This day of waiting is usually extremely painful and to get around that, I have collected and compiled data from various discord and whatsapp channels to produce this document. This report is written by someone who does not know what they are doing, so take it with a huge pinch of salt, and please do suggest better methodology. Compiling this data will be only the simplest step we can take in understanding how the entrance process really works, as very little information about it is released to the public.

Data for previous years

How the data is collected

I went into all of the TU Delft discords i had, went into the search bar, searched for "rank". This led me to spikes of when messages were sent, and these message activity spikes (usually over a few days) included a lot of screenshots of scores with ranks included. These were added to a spreadsheet `data.ods`, and used to create the following plot.

Analysis

I knew that the expected distribution would be a gaussian distribution. Ranking each member would involve finding the percentile at every point. To do this, the cumulative distribution function of the normal distribution would be used.

The cumulative distribution function is equivalent to the indefinite integral of the gaussian distribution, which is known as the error function erf.

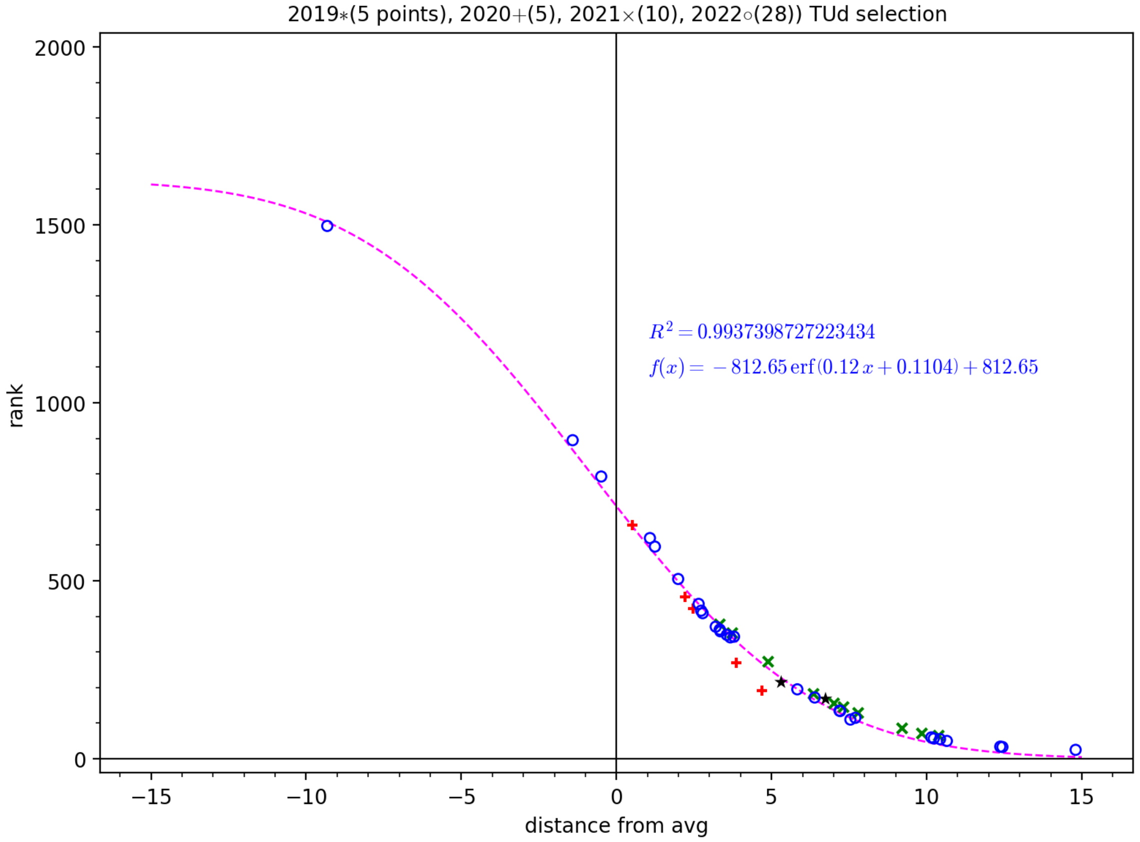
The data given to me was already ranked, so i knew the expected fit would be an erf fit.

*Special thanks to Django van der Plas and everyone on the TU Delft Study server. If at any time the word "we" is used, it refers to my conversations with various people in the server.

To find the model below, sagemath is used.

```
var("a,b,c")
erfc(x) = 1-erf(x)
model(x) = a * erfc( b * (x+c))
fit1 = find_fit(year.astype(float),model)
```

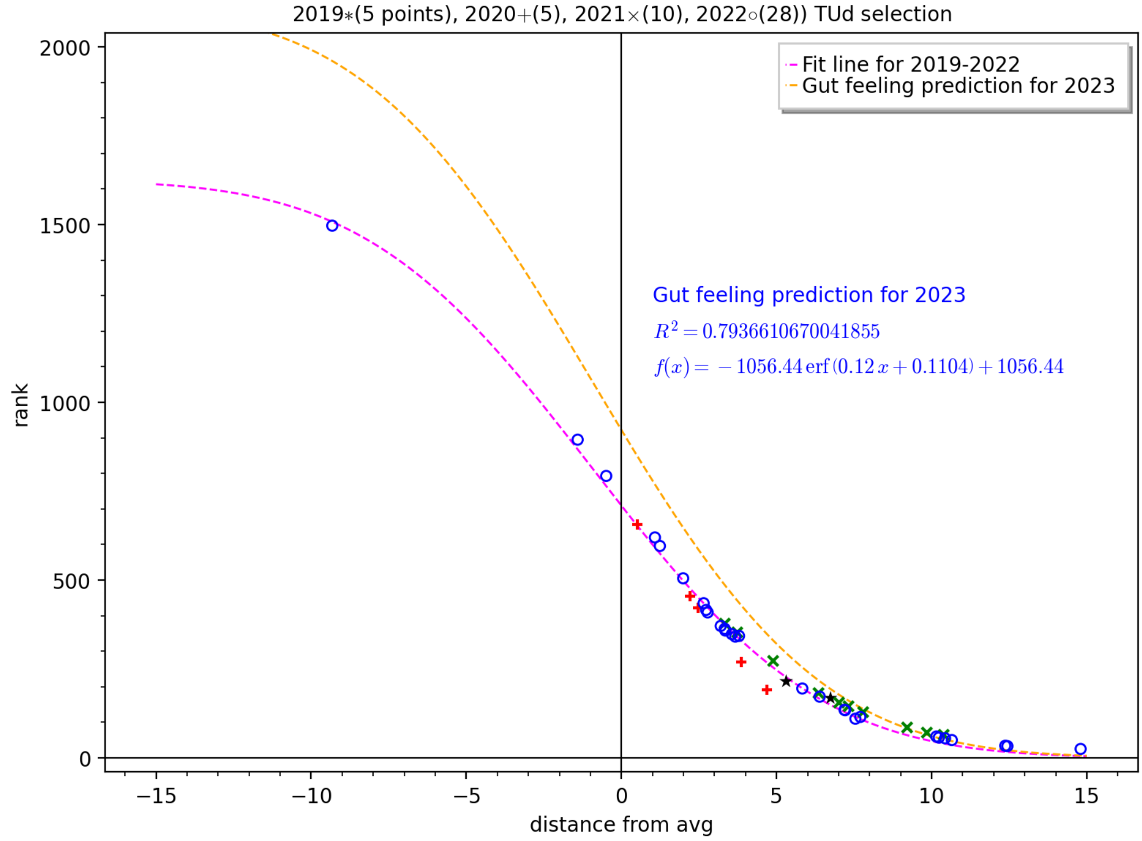
The full code can be inspected in the source of this document.



This model seemed to be accurate to within 40 ranks when tried with data not included in the training set.

However, this model will be inaccurate when applied blindly to 2023 ranks, because the lowest rank is a lot higher than 2022 due to a higher quantity of applicants. Thus, we scale the model by increasing the first coefficient.

The ceiling of the function (i.e the lowest rank) for 2022 was 1600. We had naively eyeballed that about 1.2-1.4 times as much people would apply this year (without any backing evidence). Thus, scaling the function respectively (by a factor of 1.3) would yield



However, I did not release this as I considered it to be completely baseless and unreliable, as the number 1.3 was completely arbitrary.

Data for 2023

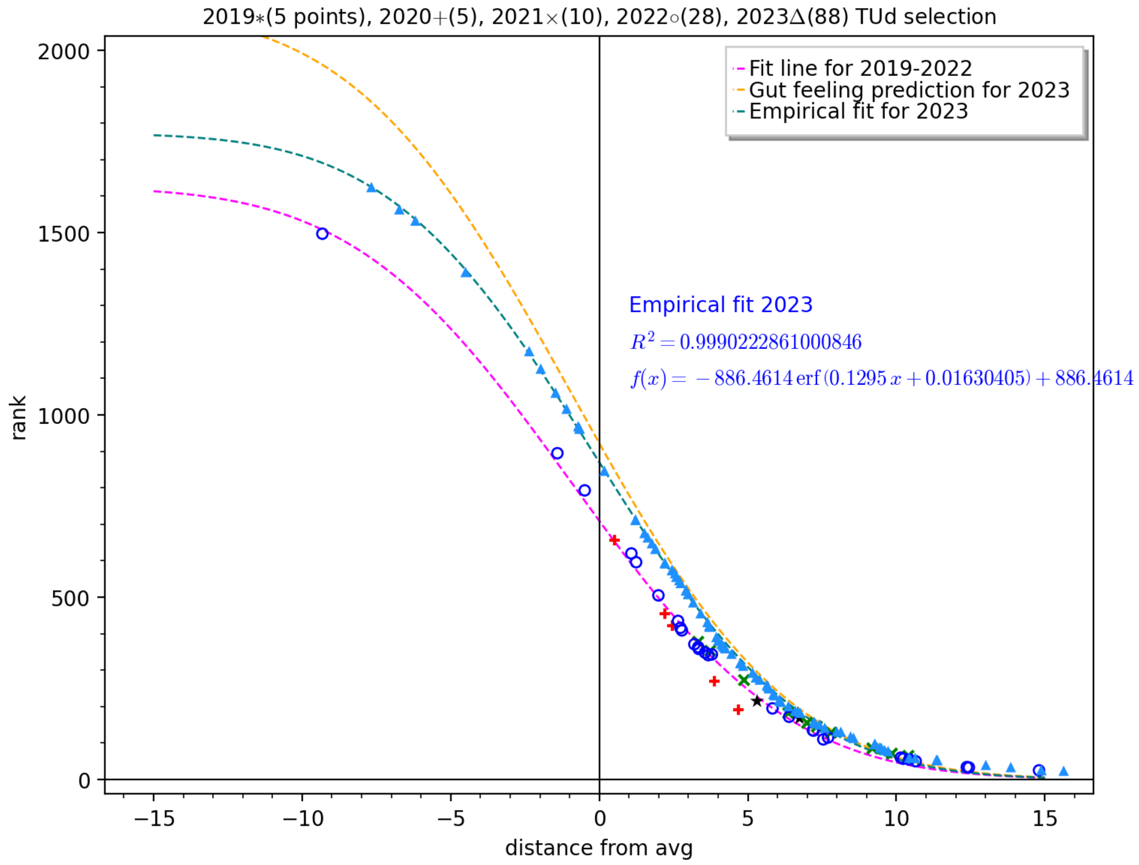
How the data was collected

The google form https://forms.gle/tzp7KgC5CznU8Q7VA was relentlessly spammed at the TU Delft discords, and also the whatsapp chat for 2023 applicants. I apologise from everyone for how annoying it must have been.

I am willing to say it paid off, because as of 2023-04-16, there are 92 responses (some of which were unusable due to trolling and/or invalid entries), which is great. Thanks to everyone who participated and donated data. I wish i could credit everyone individually. The persons who included their name will be in the thanks section.

Analysis

I didnt change the code much for the 2023 analysis from the last plot. It is mostly the same stuff.



Playing around

Now that i knew the fit line for 2023, i had a rough idea how many more people applied.

My fit for 2019-2022 yielded

```
1625.2991992030168
```

as the lowest rank.

The e-mail sent last year showed that 2300 people applied and about 1800 finished the mini-mooc. I suspect that some people dropped after that as well, however a drop of

```
174.70080079698323
```

seems a little too much.

Yet still, let us asssume this number is correct for now.

The highest rank yielded by the 2023 fit is

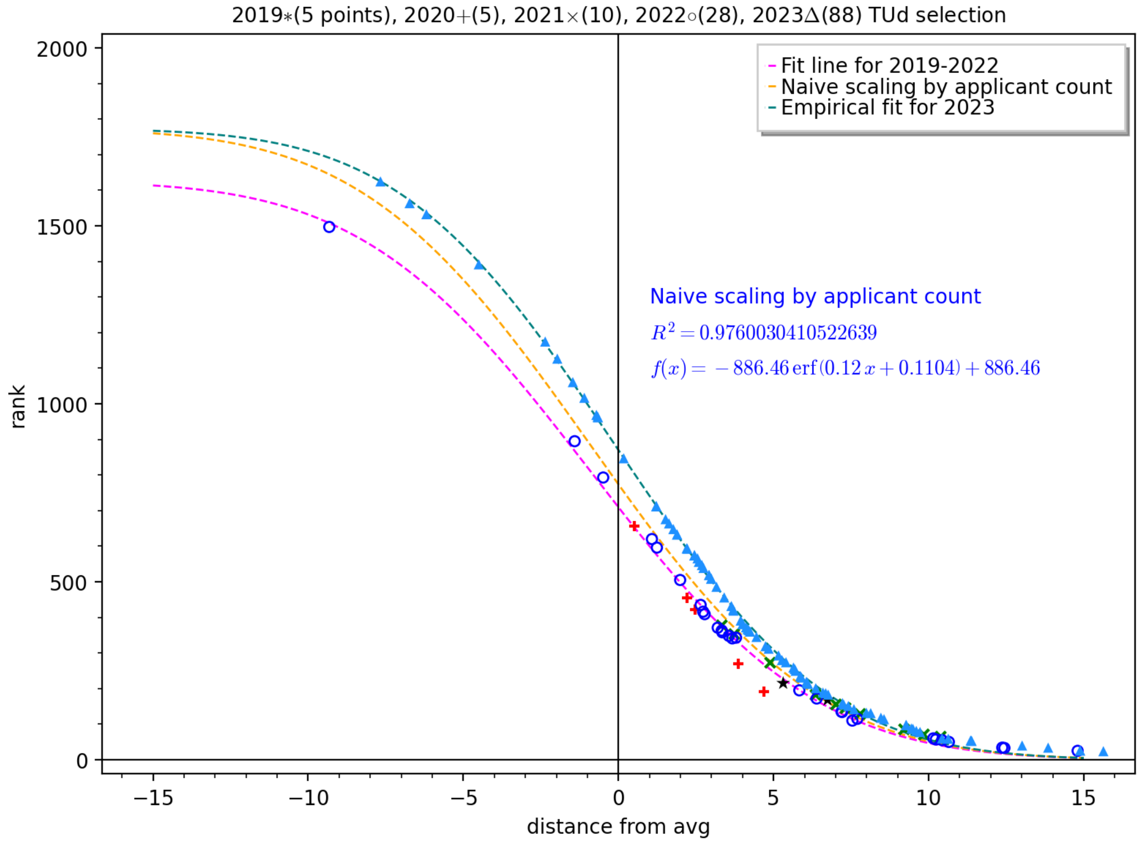
```
1772.9227959772777
```

If we are to assume this number is correct,

```
1.09082856673199
```

times as much people applied.

As you might remember, my estimation was 1.3 times. Instead if my naive prediction had used `1.09082856673199` as the coefficient, the following line would appear



Note that the minimum and maximum ranks on both the orange “Naive scaling by applicant count” and teal “Empirical fit for 2023” lines are the same, yet the slope is different. The slope on the 2023 fit line is steeper, which leads me to believe that the exam was harder this year.

At this point i’d like to remind the reader that i do not know what i am doing, and that corrections are most welcome.

Conclusion

Predicting rank using this methodology, i.e getting the last year's curve and scaling it proportionally to this year's applicant count is a naive approach and yields very inaccurate results. Add to that the fact that the applicant counts are not released, makes this method incredibly inaccurate.