Lichess Rating Prediction Models

General Information

- The Lichess rating prediction bot uses two models:
    - 1) A binary classification model to predict whether or not a user will reach a target rating for a specified time control within 2 years.
    - 2) A regression model to predict when a user is expected to reach the target rating, if they do reach it within 2 years.
- The models are both trained using a sample of ~10,500 lichess users from the Lichess Swiss team. The users have had accounts since July 2019 or earlier. The model supports bullet, blitz, rapid, and classical time controls.
    - I used an 80/20 train/test split for both the classification and the regression models. The metrics in this document describe test set performance.
- The models predict what will happen to a user's rating given the information available 2 years ago, in August 2019. The observed outcomes occur between then and August 2021, when the data was collected and the model was built. An assumption of this approach is that users of the discord bot who have similar values of the model predictors as Lichess Swiss users in August 2019 will have similar rating growth patterns to them.
- The training data includes 5 randomly generated target ratings for each user and time control. The sampling distribution depends on the user's current rating, and is intended to give the greatest sample size to rating gains between 0-300 while including some training data for 300-700 point rating gains.
- The features used in the models are:
    - target rating gain
    - current rating
    - peak rating
    - rating values at various time lags
    - rating variance

- frequency of rating updates
- specified time control
- transformations and interactions of these features.

Classification model performance

- The classifier is a binary logistic regression model
- In 70% of the test set the target rating is achieved.
- Distribution of modeled probabilities in the training data:

Minimum	25th Percentile	Mean	Median	75th Percentile	Maximum
0%	47%	71%	86%	96%	100%

- The model's test set ROC AUC is 90%. For rating gains that are harder to predict, in particular those between 50 and 200 points, the model's ROC AUC is 78%.
  - Baseline ROC AUC is always 50%
  - A logistic regression with only target rating and latest rating as predictors has a test set ROC AUC of 88% overall, and 70% on the 50-200 group
- ROC AUC by time control:

Overall	Bullet	Blitz	Rapid	Classical
90%	90%	92%	89%	86%

Regression model performance

- The regression model is Ordinary Least Squares
- Actual and Predicted Statistics in Days (*predictions outside the range of 0-730 are overridden to be 0 or 730).

	Minimum	25th	Median	Mean	75th	Maximum

		Percentile			Percentile	
Actual	1	39	130	201	295	730
Predicted	0*	112	178	202	275	719*

- Error statistics

	Minimum	25th Percentile	Median	Mean	75th Percentile	Maximum
Raw error	-694	-65	36	0	102	562
Absolute error	0	44	93	119	159	694

- Absolute error statistics by time control (days)

time_control	count	mean	std	min	25%	50%	75%	max
Blitz	6240	116	104	0	40	86	156	677
Bullet	4427	113	107	0	38	84	148	694
Classical	1291	124	102	0	55	104	158	669
Rapid	3172	133	99	0	61	113	180	616

- I compared the test set results of the OLS multiple regression model to the logarithmic fit extrapolation model from the original rating prediction bot. To ensure a fair comparison, I let the logarithmic fit model replace negative predictions with zeros just as the OLS multiple regression does. I also set the > 25 days of data requirement. It is clear from the following table that the OLS multiple regression greatly outperforms the logarithmic fit model. The mean absolute error is ~15X smaller and the median absolute error is ~1.7X smaller.

- Note: “log_error” in the following tables refers to error measured in days from the predictions of the logarithmic fit model, not a logarithmic transformation of the error

	ols_error	ols_abs_error	log_error	log_abs_error
count	9821	9821	9821	9821
mean	3	111	1467	1612
std	149	100	56655	56651
min	-694	0	-730	0
25%	-47	39	-77	43
50%	36	83	3	144
75%	96	148	248	384
max	436	694	5243715	5243715

The OLS model has lower mean and median absolute error for all time controls.

	ols_abs_error		log_abs_error	
	mean	median	mean	median
time_control				
Blitz	110	81	1440	146
Bullet	103	72	624	136
Classical	104	83	306	99
Rapid	126	107	4073	167

- However, this might be somewhat of an unfair comparison since the multiple regression model knows that the answer will never be greater than 2 years because of the scope of the training data, but the logarithmic fit model doesn’t have that insight. So I tried another version where I compared the two models on observations where the log model predicted less than 1 year - this way, the possibility of a > 2 year prediction by the logarithmic fit model is small and irrelevant. Here, the OLS multiple regression model is still more accurate, but by a smaller amount.

	ols_error	ols_abs_error	log_error	log_abs_error
count	7012	7012	7012	7012
mean	10	100	-62	136
std	138	96	192	149
min	-694	0	-730	0
25%	-25	34	-131	26
50%	38	75	-21	84
75%	92	132	39	192
max	431	694	357	730

- Besides the quantitative performance metrics, I believe the updated models have some additional advantages
    - The models acknowledge the possibility that the target rating gain won't be achieved and offer an estimate of that probability
    - The models follow a supervised learning approach so the predictions are connected to actual rating performance outcomes by real users. This means that the models can be improved in the future by including more features, a larger sample size, and more advanced classification and regression algorithms.
- Other future steps might include:
    - Adding a prediction interval in addition to a point estimate for the predicted rating date
    - Expanding the model to include other variants / time controls