Predicting winners in esports tournaments (Super Smash Bros Melee)

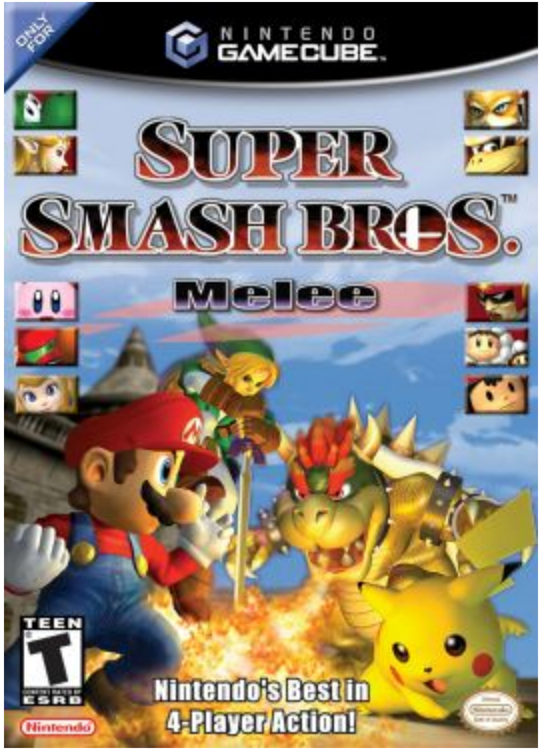
THE ERDŐS INSTITUTE
Helping PhDs get and create jobs they love at every stage of their career.

Dan Ursu
Jaspar Wiart

The setup

- Fighting game released in 2001
- Tournaments with tens of thousands in prizes
- Viewership in the hundreds of thousands

The goal is simple: Predict the winner



The dataset

Courtesy of smashdata.gg, have tournament data from 2015 onwards on github

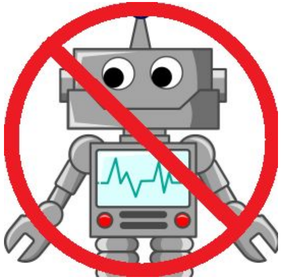
Includes 1,800,000 **sets** played in 39,000 **tournaments** between 96,000 **players**.

Pros:
- Large dataset
- Easy to obtain

Cons:
- Large dataset
- Missing values, few ready features

Therefore, **more**:
- Feature engineering
- Efficient code

and, **less**:
- Super fancy AI



The baseline

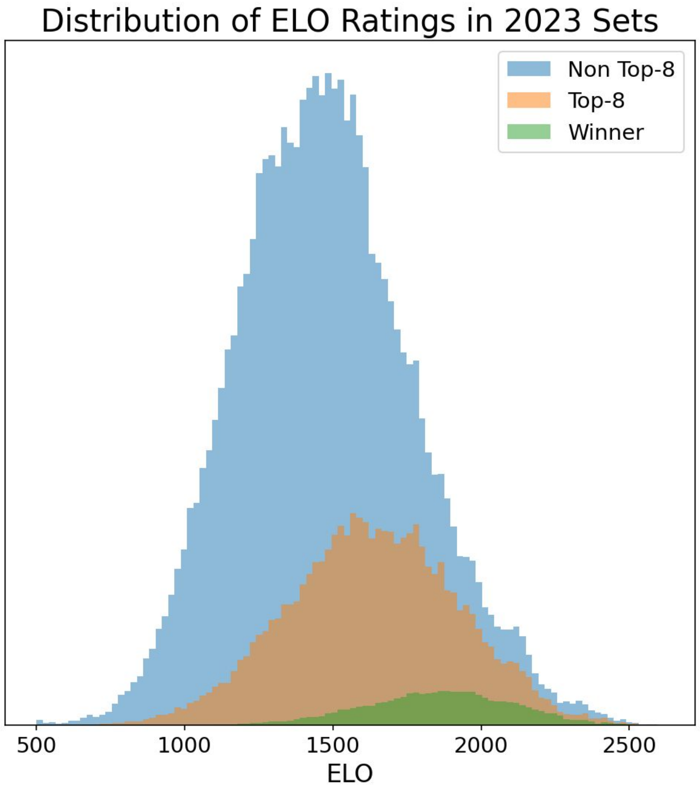
Sports typically have Elo or Elo-like score keep track of player skill levels over time.

Currently popular: **Glicko-2** (c.f. Wikipedia)

Baseline model: “whoever has the highest Elo”

Note: Glicko-2 is quite sophisticated, and predicting sports outcomes is **hard**.

Any small improvement on baseline is a success.



Engineered features

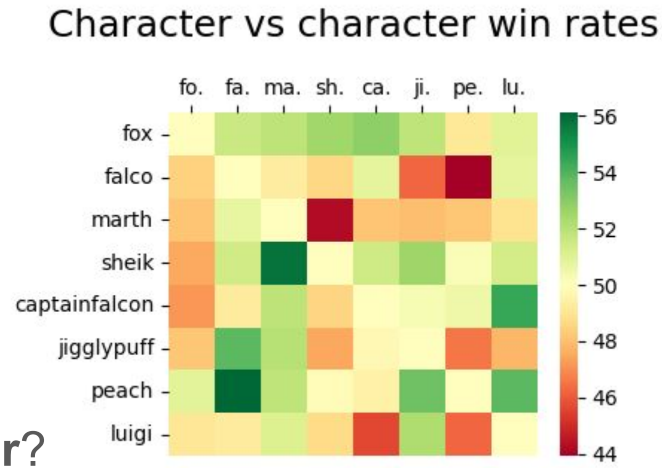
In Super Smash Bros, players choose **characters** to fight with.

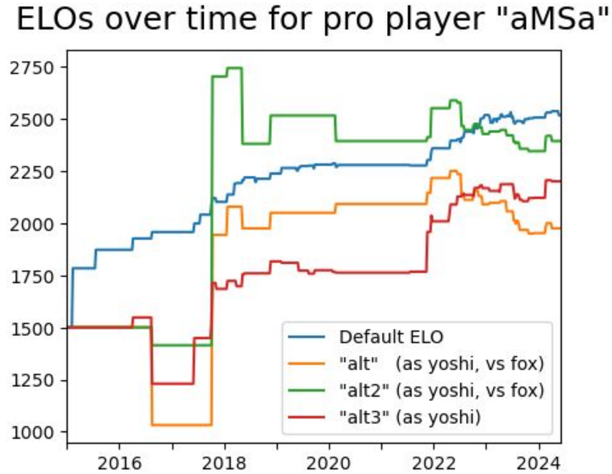
Perhaps some players do better depending on the opponent's **character**?

Most important engineered features:

Modified "Elo" scores that take into account characters played.

Three variations: "alt", "alt2", "alt3".





A model for individual matches

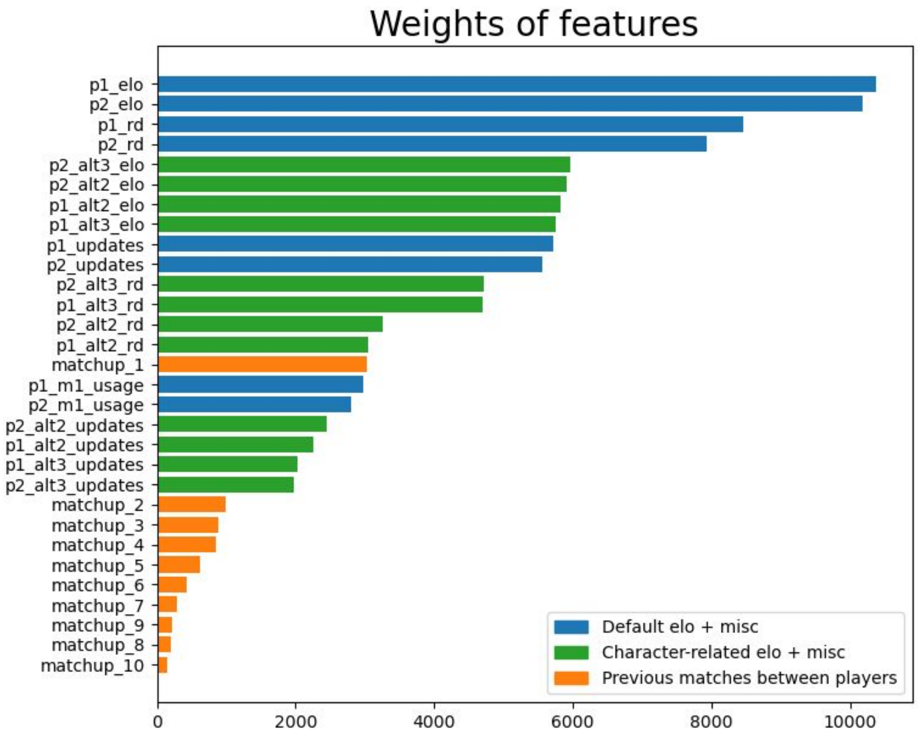
Model	Accuracy (all matches)	Accuracy (top 8 matches)
"Who has the highest Elo"	77.56 ± 0.16	73.89 ± 0.36
XGBoost on default Elo only	79.05 ± 0.16	74.04 ± 0.36
XGBoost on all engineered features	79.89 ± 0.16	75.03 ± 0.35

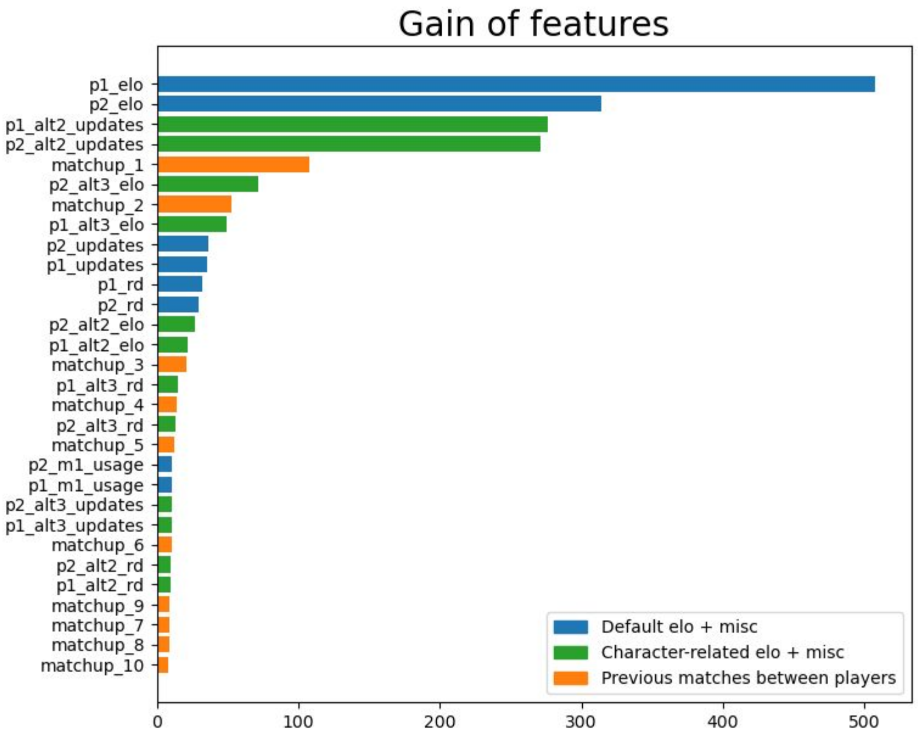
(with 95% confidence intervals)

Some observations:

- A definitive increase in accuracy of about 1% 🥳
- Accuracy on top 8 sets is decreased (substantially lower skill difference)

Feature importance





The graveyard of failed ideas, I

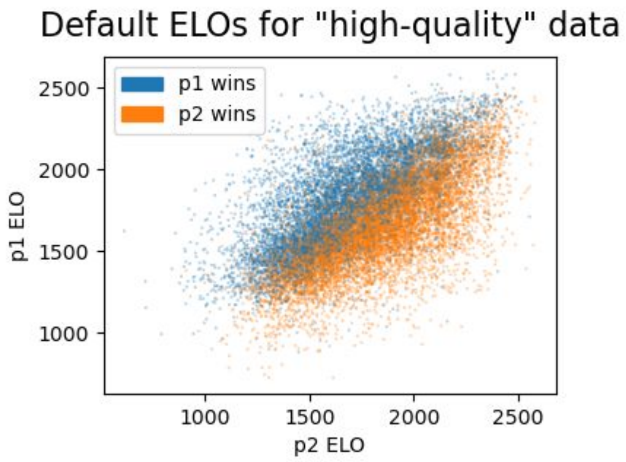
Linear models for subsets of data:

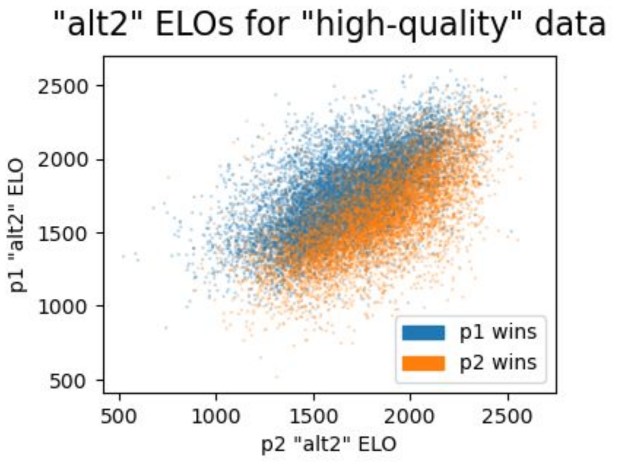
“High-quality” data followed multivariate normal distributions.

- Tried splitting off this data and applying logistic regression or LDA

- Tried rolling our own errors-in-variables version of LDA

Underperformed XGBoost





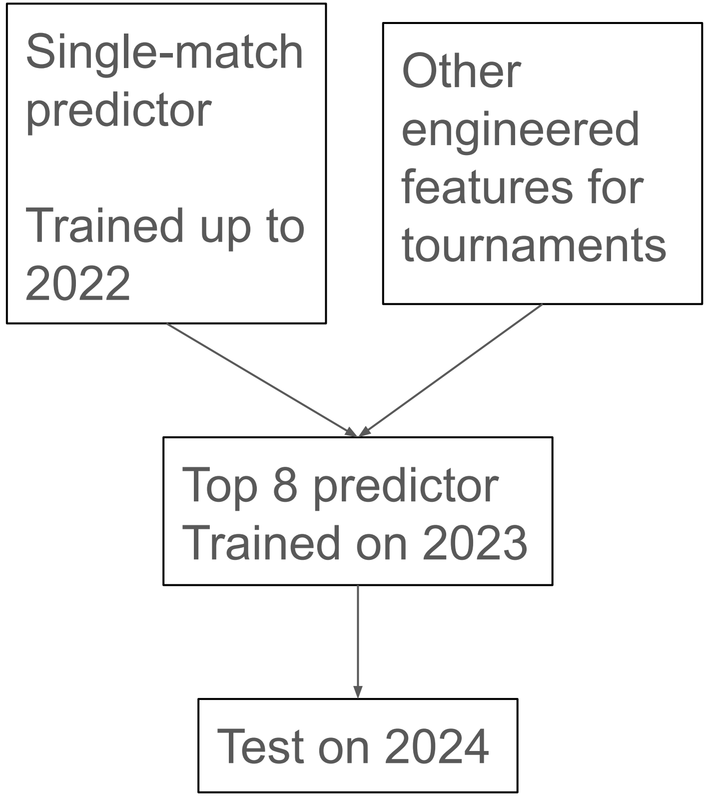
The graveyard of failed ideas, II

Predicting the winner of the top 8 finalists:

Computed pairwise probabilities using single-match model and go from there.

- Tried feeding these + pre-top-8 performance data into XGBoost.
- Tried simulating all ways top 8 could play out.

Did not outperform baseline (70.2 ± 1.3)



In summary

Conclusion:

- Engineered modified Elo variants that take into account characters
- Model trained on all engineered features performed better than just using default Elo

Future work:

- Trying other, more sophisticated models (neural nets, etc…)
- Seeing if top 8 predictor can be used for predicting upsets and other tasks
- Seeing if engineered features are applicable to other esports

- *Fin* -