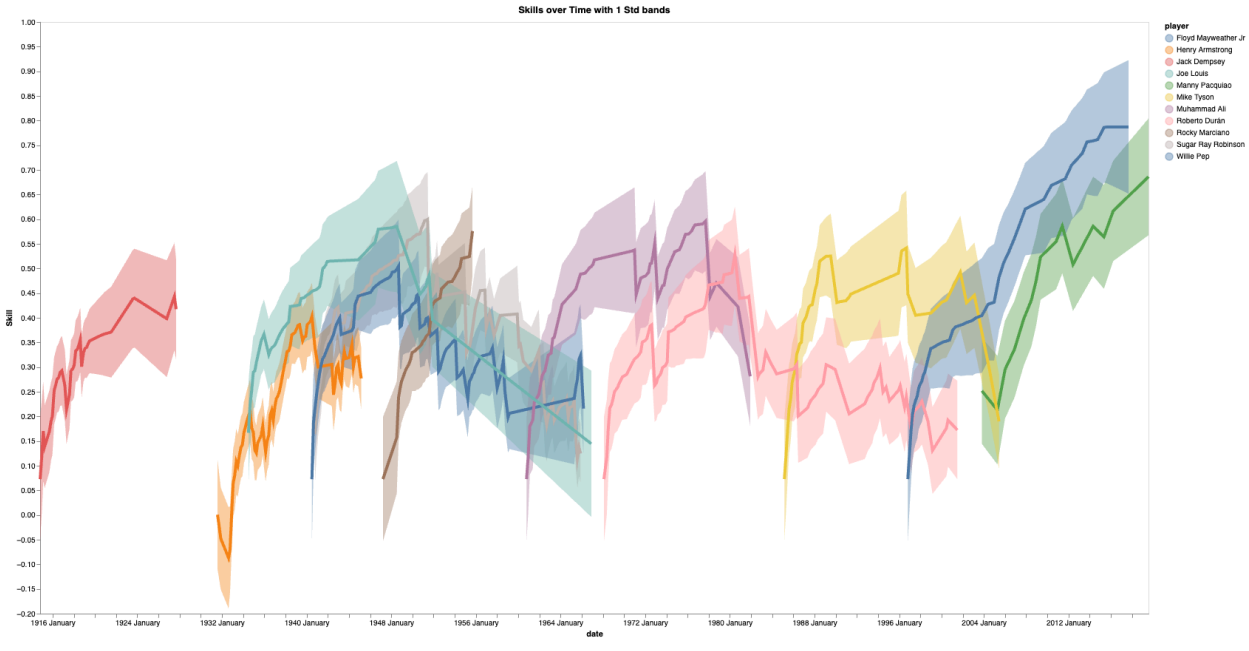
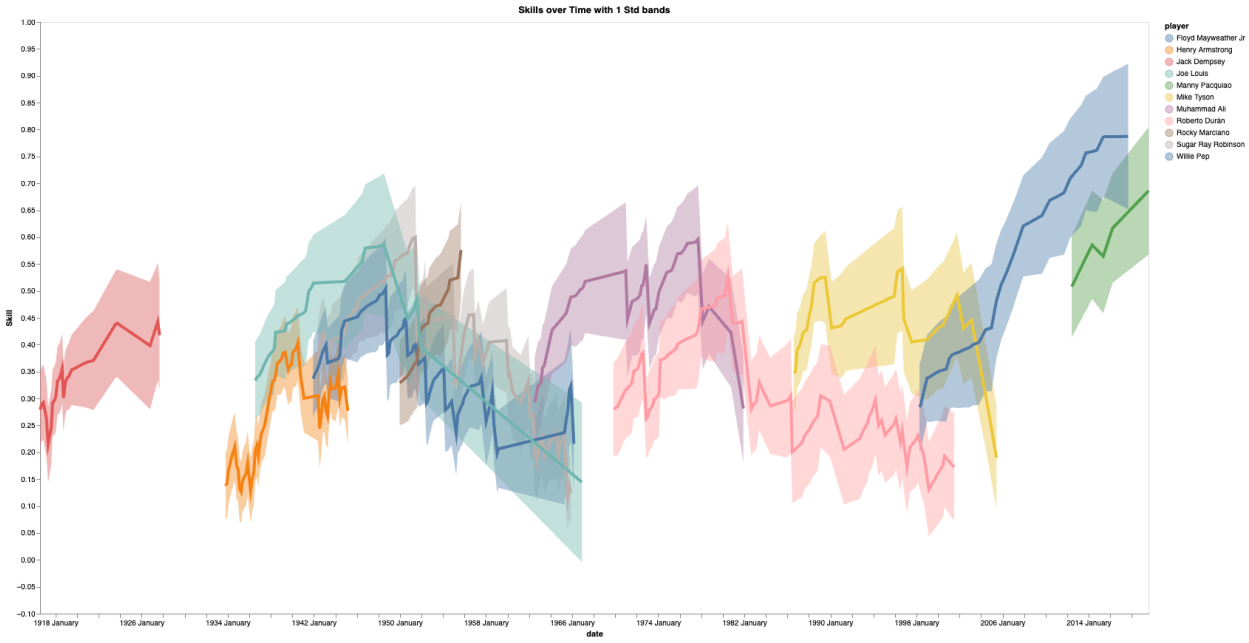
Apply TrueSkillThroughTime algo on the full dataset(25651 matches):

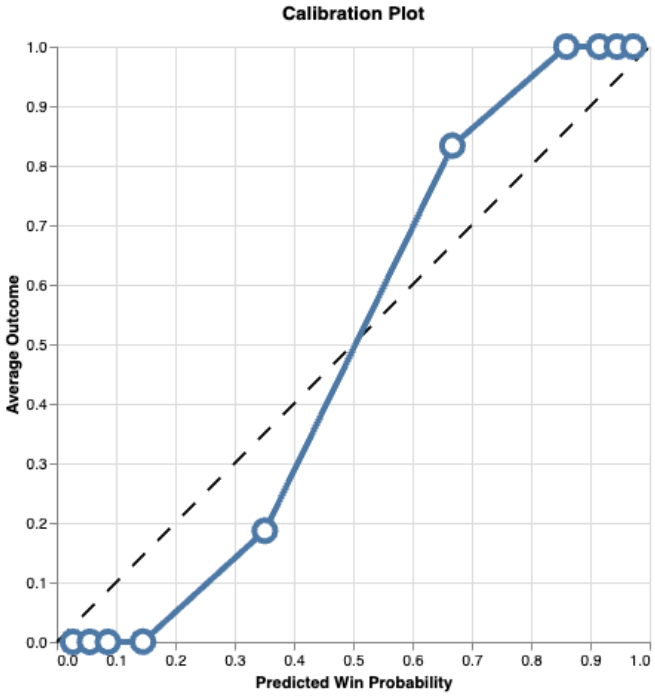
```
top_boxers_online = ['Muhammad Ali', 'Joe Louis', 'Sugar Ray Robinson', 'Rocky Marciano', 'Floyd Mayweather Jr', 'Manny Pacquiao', 'Jack Dempsey', 'Roberto Durán', 'Henry Armstrong', 'Willie Pep', 'Mike Tyson']

self.plot_player_skills(players = top_boxers_online, width=1500, burnin=0)
```



```
self.plot_player_skills(players = top_boxers_online, width=1500, burnin=10)
```




```
self.plot_calibration()
```

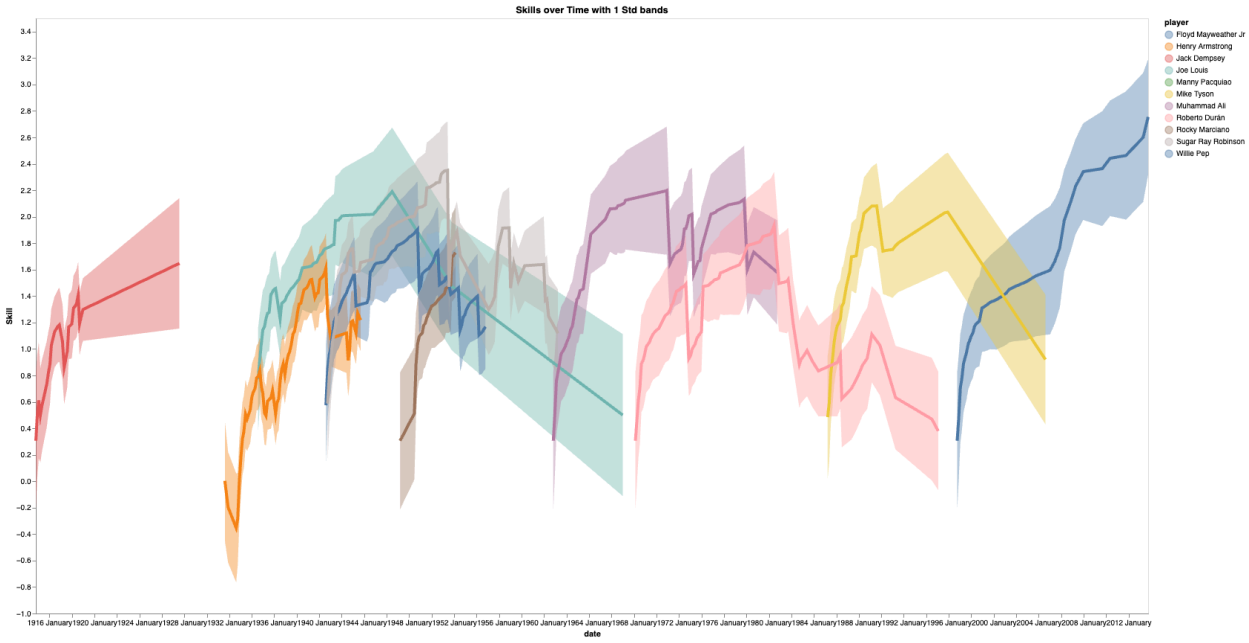


Out-of-sample evaluation:

- We do a 80:20 train test split based on time for each one of 303 players who has played at least 40 matches in their career.
    - data needs to be converted back to the winner-loser-timestamp format in order to apply TrueSkillThroughTime algo.
- We use the first 80% of a player's match to train the model and evaluate his last 20% of matches in the data.
    - (see posts/trueskill/data/oos_eval/games_ge_40_train.csv and posts/trueskill/data/oos_eval/games_ge_40_test.csv)
- To evaluate the algo, for each player in the training set, we take his last point in the skill_curve and use the mu at that point to measure the boxer's performance.
    - It turns out that the accuracy is **73.21%**(i.e. there are 73.21% of matches in the test say whose winner have a higher mu than the loser)
    - We predict the boxer would win the match if his last mu is greater than his opponent's last mu.
    - We make sure there is no overlap between the train and test set. For example game_index 13416, it's Aaron Pryor,1982-11-12,13416,1 in train.csv, however it's Alexis Argüello,1982-11-12,13416,0 in test.csv; It is Aaron Pryor's first 80% of matches in his career however it's Alexis Argüello last 20% of his career.
    - There are only 5% of such overlapping matches in the test set. (258 out of 4387 matches)
    - And players not on the 303 player list will have a mu of zero as assumption.

Do a train-test split of data:
Filter only those players who have played at least 40 matches, then for each player, do a 80:20 split by timestamp and apply the algo on the training set(of course the data needs to be converted back to the winner-loser-timestamp format).

```
self.plot_player_skills(players = top_boxers_online[:]+['Mike Tyson'], width=1500,
burnin=0)
```



The overall trend for the train set is similar to the result of the full sample, however players on the train set have higher skill levels in absolute value.

For the entire dataset, here is the statistic that a boxer has played.

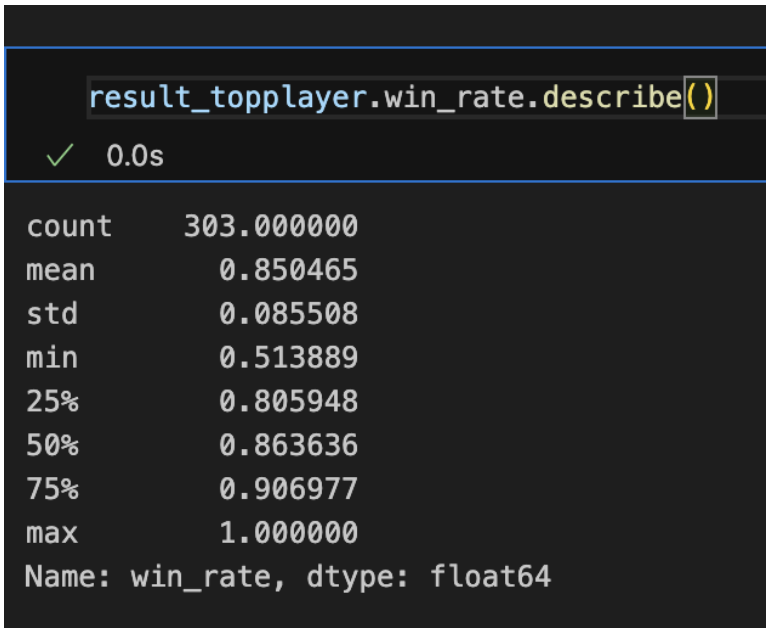
```
total_matches_df = games.winner.value_counts().add(games.loser.value_counts(), fill_value=0).sort_values()
```

```
total_matches_df.describe()
```

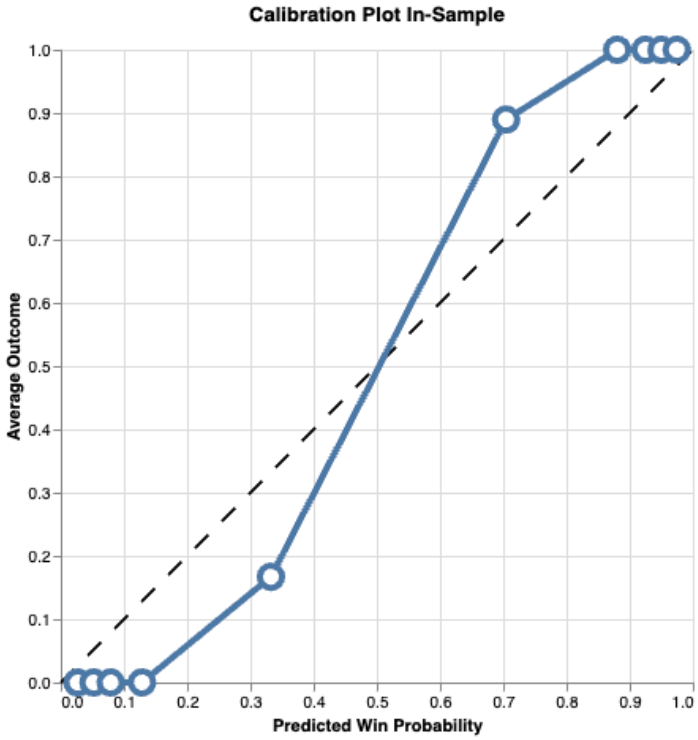
```
count    16033.000000
mean         3.199775
std         11.276392
min          1.000000
25%          1.000000
50%          1.000000
75%          2.000000
max        280.000000
Name: count, dtype: float64
```

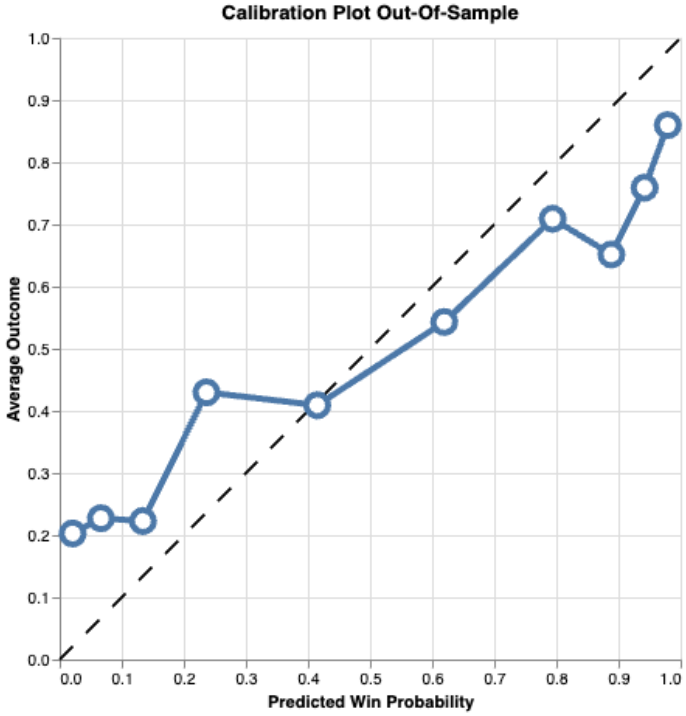
Even after we focus on boxers who have played over 40 matches, we still have to include many boxers who have played few because they are the opponents of those frequently played boxers.

There are 303 players who have played at least 40 matches in their career. The win rate for those players are quite high, which makes boxing a very winner-dominant game.



In-sample vs Out-of-sample calibration plot:

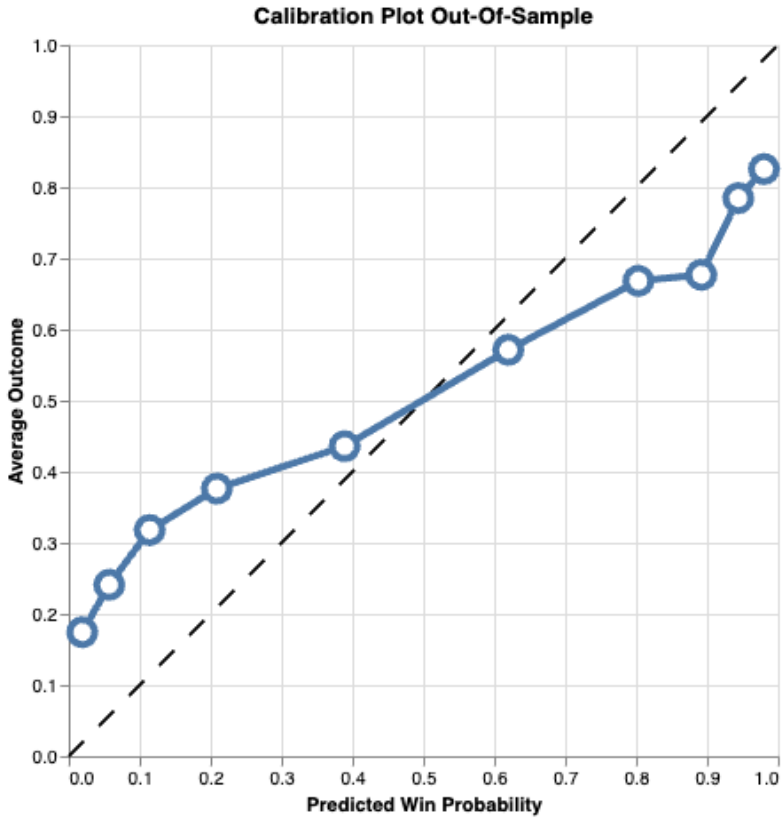




As in the Out-Of-Sample calibration plot, we have a closer to diagonal line for pred_win_prob vs avg_outcome compared to the plot of the training set. However, the model tends to

underestimate the underdog player and overestimate the top dog since it only uses the historical data to calculate win_prob.

After we filter out those matches that appear on both train and test set(about 200 such matches), we do the OOS calibration plot again.



roc_auc_score:
Assign a match to positive if the winner is alphabetically lower than the loser and is negative otherwise. The dataset is relatively equally balanced over the classes.
If we evaluate on the entire dataset:
auc: 0.9965

If we evaluate on the 20% test dataset:
auc: 0.7421

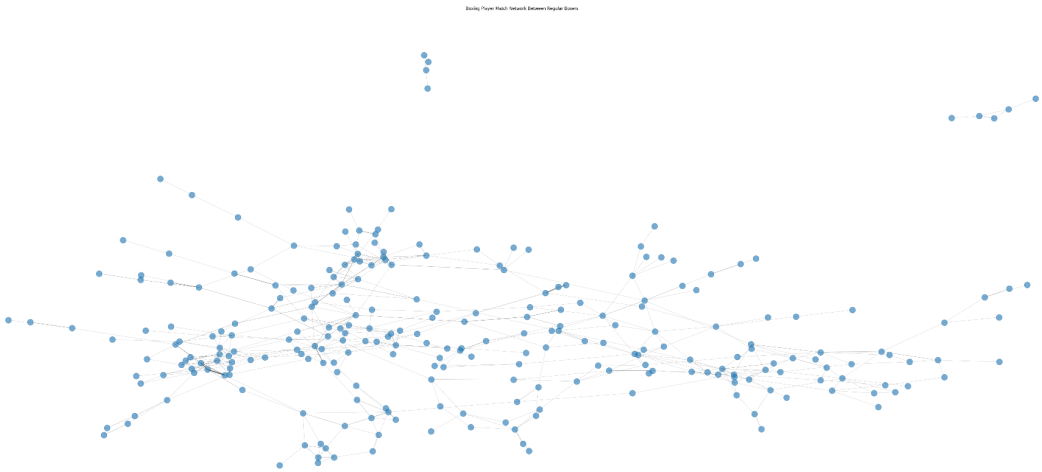
Auc for the 10K OOS Warcraft 3 dataset is 0.6336.
The explanation is that the Warcraft 3 OOS gives about 400K games after the transformation, compared to only about 3K games in boxing OOS.
And Warcraft 3 players in general play more frequently and their skills change faster through time.

Game Network Graph:
Draw a game graph with at least one player who played over 40 games in their career. The nodes are players and edges are if they have played against each other, the edge weight is the count of games between each node. We consider those 303 players regular players.

We use the entire data of 25651 games. After the filtering on regular players, there are 20711 games left.
There are 13127 nodes in the graph, and there are 29 clusters(communities) detected.

Filtering out games with both coming from less regular players, there are more clusters on the boxing graph than in warcraft3, meaning that warcraft3 players play more often with each other. In this way, trueskill algo is more capable of making predictions of the skills based on historical game results.

If we only focus on games played between regular players, there are 622 games left.
The graph has 265 nodes, and there are 16 clusters(communities) detected.



There are 16 groups detected by community.community_louvain algo, and the center(the node with the max centrality) of each group is highlighted in bold. It turns out that many of the centers are the candidates of the GOAT(Floyd Mayweather Jr, Muhammad Ali, Joe Louis, Sugar Ray Robinson, Roberto Durán etc.).
Those nodes are somewhat classified by the weight class. If you look at nodes in the groups of Muhammad Ali for example, many of the players are heavyweight class. And if you look at those of Floyd Mayweather Jr, many of them are lightweight or middleweight. By the way, Floyd Mayweather Jr has played for five different weight classes so his opponents have wider varieties in the weight class than Muhammad Ali who has only played heavyweight class.

(Need to zoom in the plot to see it better)