How to Make Money with Machine Learning on Sport Betting

Utilizing Machine Learning on Horse Racing Betting Strategy

Andrewngai Follow
Feb 21 · 8 min read ★

Note from the editor: *This article is for educational and entertainment purposes only. If you want to use the presented model for real money bets, you do it at your own risk. Please make sure that it is in alignment with the terms and conditions of your bookmaker.*

Machine learning has been widely used in many time series analysis and forecasting. With the help of a large amount of historical data and computing power nowadays, ML models can sometimes produce extremely useful insight and guidance to sports betting decision making.

Photo by Julia Joppien on Unsplash

This article illustrates how machine learning could help with horse racing betting strategy. We will use the data crawled from the Hong Kong Jockey

Club home page, one of the oldest and largest horse racing institutes in the world. To avoid data leakage and evaluate the real performance of the model, we will be only using the matching data from the beginning of 2007 to fall 2019 to build the model and use it to bet on new upcoming matches. We've utilized the model and build a unique investment strategy to bet for a two-month period(2019/09–2019/11) and achieve a positive return in the experiment.

Dataset

As we mentioned earlier, we will be using all the games from 2007 to 2019 in Hong Kong as training and validation sets. And 2019 winter data for test set to evaluate the overall betting portfolio performance. There are 109085 rows and 61 columns in the training data containing various information about each game.

Columns

index
Rdate: Race date
Rid: Race ID

Hid: Horse ID

Venue: Race venue (HV, ST)

Track: Race track (TURF, AWT)

Going: Race track condition

Course: Race course (AWT will have no specific race course description)

Class: Race class of the match

Distance

Rfinishm: Race finish time in centi second (1/100 second)

Rm1: 1st section finish time of the race in centi second

Rm2: 2nd section finish time of the race in centi second

Rm3: 3rd section finish time of the race in centi second

Rm4: 4th section finish time of the race in centi second

Rm5: 5th section finish time of the race in centi second

Rm6: 6th section finish time of the race in centi second

Horsenum: the horse ID of the horse

Jname: Jockey

Tname: Trainer

Exweight: the handicapped weight carried by the horse

Bardraw: draw

Gear: Gear putting on the horse

Rating: Horse Rating

Ratechg: Rating change of the horse from previous race

Horseweight: Horse's weight
Horseweightchg: The Horse's weight change from previous race
Besttime: The best finishing time of the horse on the race with same venue, distance and track (minute. second. centi second)
Age: age of horse
Priority: Priority of the horse for race given by the trainer.
Lastsix: the rank from previous 6 races
Rank: rank in current match
Runpos: The rank in each section of the horse in the race.
P1: The rank in 1st section of the horse
P2: The rank in 2nd section of the horse
P3: The rank in 3rd section of the horse
P4: The rank in 4th section of the horse
P5: The rank in 5th section of the horse
P6: The rank in 6th section of the horse
M1: The finish time of the 1st section of the horse in centi second
M2: The finish time of the 2nd section of the horse in centi second
M3: The finish time of the 3rd section of the horse in centi second
M4: The finish time of the 4th section of the horse in centi second
M5: The finish time of the 5th section of the horse in centi second
M6: The finish time of the 6th section of the horse in centi second
Finishm: The race finish time of the horse in centi second

D1: The distance from the rank 1 horse in the 1st section (0.25 meant the distance is within 1 horse distance)
D2: The distance from the rank 1 horse in the 2nd section (0.25 meant the distance is within 1 horse distance)
D3: The distance from the rank 1 horse in the 3rd section (0.25 meant the distance is within 1 horse distance)
D4: The distance from the rank 1 horse in the 4th section (0.25 meant the distance is within 1 horse distance)
D5: The distance from the rank 1 horse in the 5th section (0.25 meant the distance is within 1 horse distance)
D6: The distance from the rank 1 horse in the 6th section (0.25 meant the distance is within 1 horse distance)
Datediff: the date difference between the previous match and current match of the horse
Pricemoney: price money of the race
Windist: The distance from the Rank1 horse.
Win_t5: win odds of the horse 5 minutes before the race
Win: final win odds of the horse
Place_t5: place odds of the horse 5 minutes before the race
Place: final place odds of the horse
Ind_win: indicator of the winner of the race (1 for winner, 0 otherwise)
Ind_pla: indicator of top 3 places in the race (1 for top 3, 0 otherwise)

Corr Heatmap of features

Feature Engineering & Modeling

The original data comes with a lot of information, we need to filter out which of them are useful and also try building new features from the data to help predicting the results. I will not provide too much detail on feature engineering, but here are some **key insights** if you would like to try by yourself.

- Horse age, draw and odds of the horse 5 min before the race have a weak correlation to the winning probability.
- New features generated from past performance(eg: last 5 match performance, past odd, total win in last 180 days, finish time, etc) could be relatively useful.
- External data like weather, temperature, horse origin and information on the jockey would increase the performance of tree-base models.
- Building different binary classification models to predict winning first place probably and winning the top 3 places produce better results.
- Model stacking (NN, XGBT, GBRT, Linear, etc) significantly improves performance.
- prediction result(winning probably) should be adjusted and normalized based on other horses in the same match.
- Perform target encoding on horse and jockey largely improve model performance.
- As a time series type problem, only use time-based cross-validation to validate performance and tune parameters.

Betting Strategy

After building a relatively useful model prediction the top 1 and top 3 winning probability of each race. I've spent a lot of time experimenting and researching on how to achieve positive returns from the models. Horse racing has a lot of uncertainties and human effort to remove any potential unfair advantages. The betting strategy becomes extremely important. After many experiments on running models against real matches, I've come up with a strategy with three essential concepts.

- **Expectation Return Ratio**
- **Lowest Risk Betting**
- **Kelly Criterion**

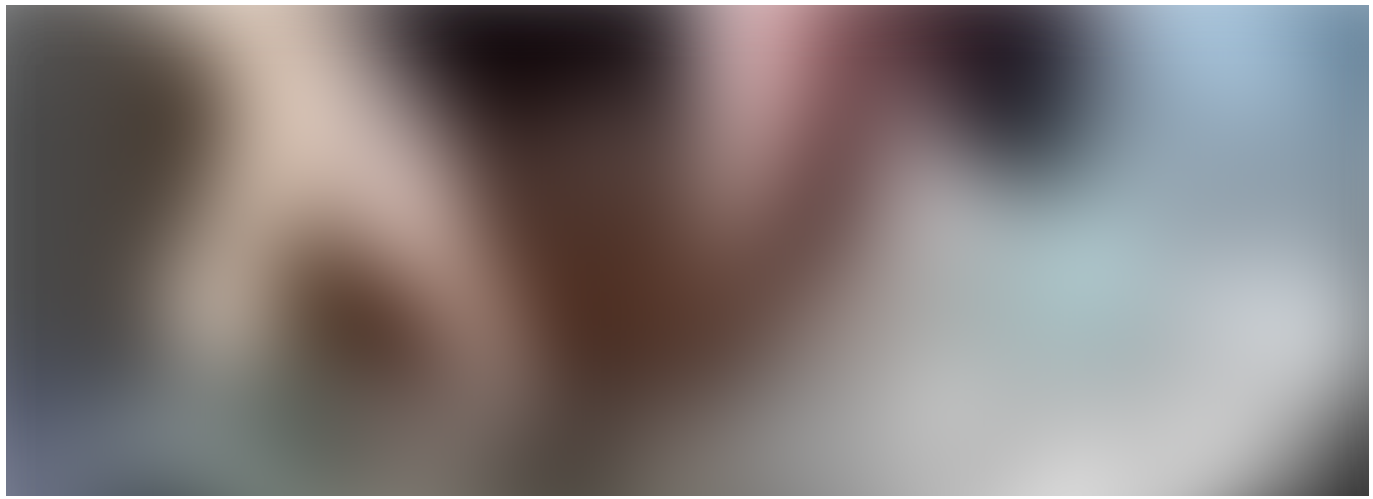

Photo by Alexander Mils on Unsplash

First, let's talk about the expected return ratio. The most common simplest betting strategy is setting a return threshold, and only bet when the return ratio(win odd * winning probability) is higher than the threshold. We just need to calculate the winning probability and return ratio of each horse on a single match using model prediction results and bet when the return ratio is higher than the threshold. However, how to choose the threshold becomes a large problem, a low threshold usually leads to aggressive betting and large gain/loss on capital. Due to the large uncertainty of horse racing, the result on both low and high threshold varies a lot.

Only using the return ratio is not good enough, as a racing game we also need to consider each horse's performance comparing with other horses in the same race. In other words, we need to find a horse that has the highest

chance of winning comparing with all other horses in the same game. To extend this concept, we could also find the horses with the highest chance winning not only in a single game but among all the matches in a single day. And we only bet on those horses to largely reduce the risk. I called this lowest risk betting. you could find those horses by building another model on the **log transformed** sum observation of the original prediction result.

Now we have all the low-risk horse and their return ratio, how much money should we bet on each horse. It turns out that, Kelly Criterion produces the best result.

Kelly Criterion Formula

For simple bets with two outcomes, one involving losing the entire amount bet, and the other involving winning the bet amount multiplied by the

payoff odds, the Kelly bet is:

- **f** is the fraction of the current bankroll to wager; (i.e. how much to bet, expressed in fraction)
- **b** is the net fractional odds received on the wager; (e.g. betting $10, on win, rewards $4 plus wager; then b=0.4)
- **p** is the probability of a win
- **1-p** is the probability of a loss

Finally, we combine those three concepts together. First, filter out all the low-risk horses of the day, and calculate their return ratio. Based on simulated past investment results to set an optimal return threshold. When the return ratio is higher than the threshold, using the Kelly Criterion to determine what percentage of the fund should bet on.

We took a two month period and apply the finalized model and betting strategy on real games. And the result is quite satisfying, we've made a bet on 76% of all the games and achieve a positive return at the end of the two month period. I will not expose the detail of the implementation, but if you have any questions or interested in my findings, feel free to leave a message below.

Thanks for reading and I am looking forward to hearing your questions and thoughts. If you want to learn more about Data Science and Cloud Computing, you can find me on Linkedin.

. . .

Photo by Alfons Morales on Unsplash