LYU2003

Applying Reinforcement Learning to “Play” Horse Racing

Kam Fung CHEUNG, Ka Yui Or
supervised by Prof. Michael Lyu

Outline

- Introduction (Motivation, Background, Objective)
- Data of Horse Racing (Collection, Description, Analysis, Preprocess)
- XGBoost (Reason using it, Progress, Result)
- Reinforcement Learning (A simple game, Horse Racing)
- Conclusion (Discussion, Problem, Future)

Introduction (Motivation)

- Reinforcement learning becomes popular in different areas especially gaming
- Horse racing and the related gambling is very popular and famous in Hong Kong
- Combine them to find out more ways to apply reinforcement learning and may make great profit

Introduction (Background)

- Reinforcement learning is one kind of machine learning and it is about how the agent learns to take actions in the environment to get the maximum reward
- At this moment, we focus on choosing the winning horse in the race
- Using reinforcement learning in horse racing means that the agent learns how to bet in horse racing to get the largest profit at the end

Introduction (Objective)

- Our objective is to build a model to place the bet on the winning horse
- The model gambles like a human. This means it can know when to bet and not to bet
- Then, we want the model to gamble better than human beings

Data (Collection)

- Use Python Beautifulsoup to do web scraping
- The horse racing data is collected from Hong Kong Jockey Club
- The weather data comes from Hong Kong Observatory

Data (Description)

- 3890 race records from 2014 to 2020 (End of 2019/2020 season)
- 2771 horse data
- 3890 weather data for each race day
- 47614 dataset for each horse in each race

Data (Description)

- Race data

Features	Description	Type	Values
race_date	The date of the race	Index	/
race_no	The numer of a race in a day	Index	/
race_index	Unique id of the race	Index	/
location	Location of the race	Categorical	HV, ST
class	Class of the horses	Categorical	Class 1 to 5, Group 1 to 3
race_length	Distance of the race	Categorical	1000, 1200, 1400, 1600, 1650, 1800, 2000, 2200, 2400
course	Track of the race	Categorical	A, A+3, B, B+2, C, C+3
draw	Draw of the horse in a race	Categorical	14 distinct values
going	Condition of the track	Categorical	FAST, SLOW, WET FAST, WET SLOW, FIRM, GOOD TO FIRM, GOOD, GOOD TO YIELDING, YIELDING,
horse_id	Unique id of the horse	Categorical	2744 distinct values
jockey_name	Unique id of jockey	Categorical	113 distinct values
trainer_name	Unique id of trainer	Categorical	112 distinct values
actual_weight	Weight added to the horse	Real value	/
declared_horse_weight	Weight of the horse	Real value	/
win_odds	The odds of betting the horse	Real value	/
place	The final place of the horse in a race	Categorical	14 distinct values
finish_time_sec	Finishing time of the horse in a race	Real value	(Seconds)

Data (Description)

- Horse data

Features	Description	Type	Values
last_actual_weight	The actual weight of last race	Real Value	/
last_declared_horse_weight	The last weight in last race	Real Value	/
diff_actual_weight	Difference actual weight between present race and last race	Real Value	/
diff_declared_horse_weight	Difference declared weight between present race and last race	Real Value	/
country	the country of the horse	Categorical	US,AUS,etc.
age	The age of the horse	Real Value	/
colour	The colour of the horse	Categorical	Bay, Chestnut, etc.
sex	The sex of the horse	Categorical	Gelding
import_type	The import type of the horse	Categorical	PP,PPG
sire_name	The name of the horse's sire	Categorical	Acclamation, Patagan,etc.
last_plc	The place in the last race	Index	/
last_rating	The rating in the last race	Index	/
rating	The rating now	Index	/

Data (Description)

- Weather data

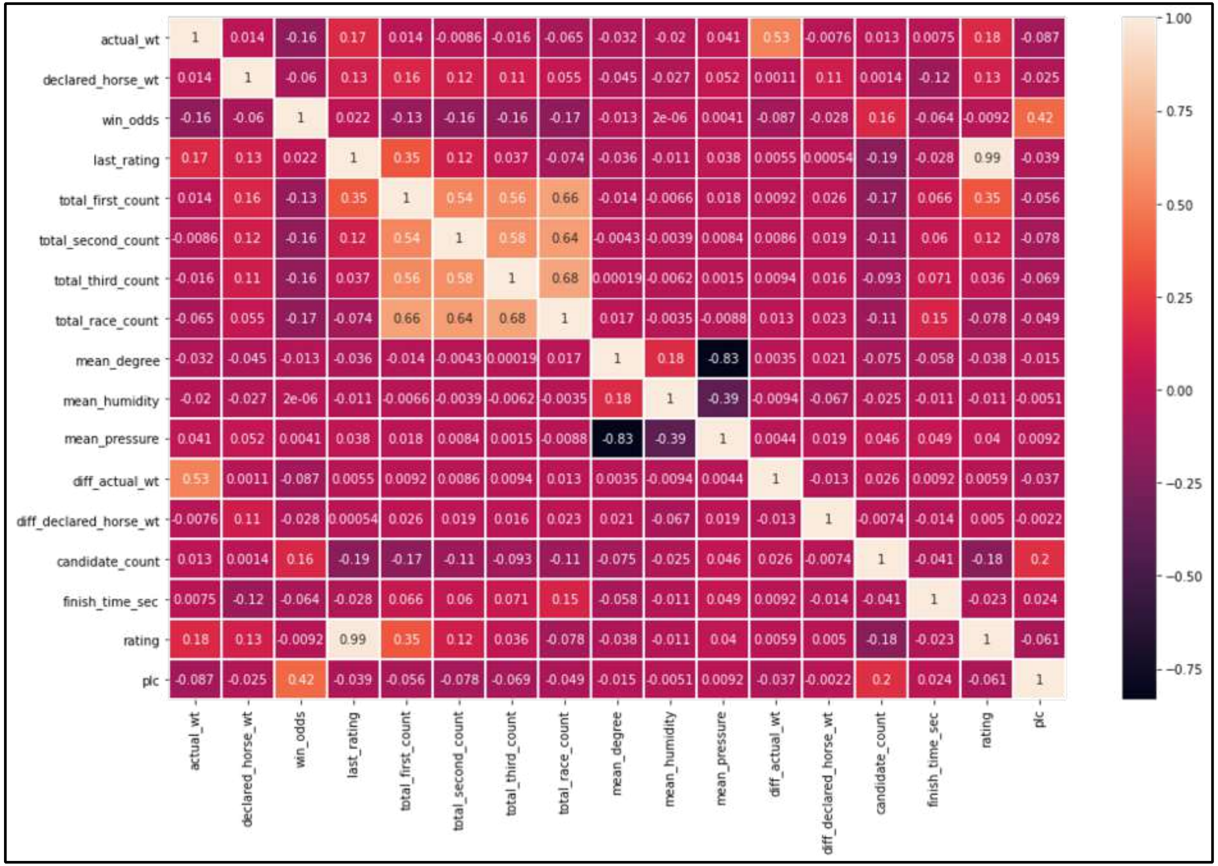
Features	Description	Type	Values
mean_degree	The mean of the temperature of the race day	Real Value	/
mean_humidity	The mean of the humidity of the race day	Real Value	/
mean_pressure	The mean of the air pressure of the race dat	Real Value	/

- Additional data

total_first_count	The total count of first place	Real Value	/
total_second_count	The total count of second place	Real Value	/
total_third_count	The total count of the third place	Real Value	/
total_race_count	The total count of joined race	Real Value	/

Data (Analysis)

- Correlation between continuous data
- Show whether the data is valid



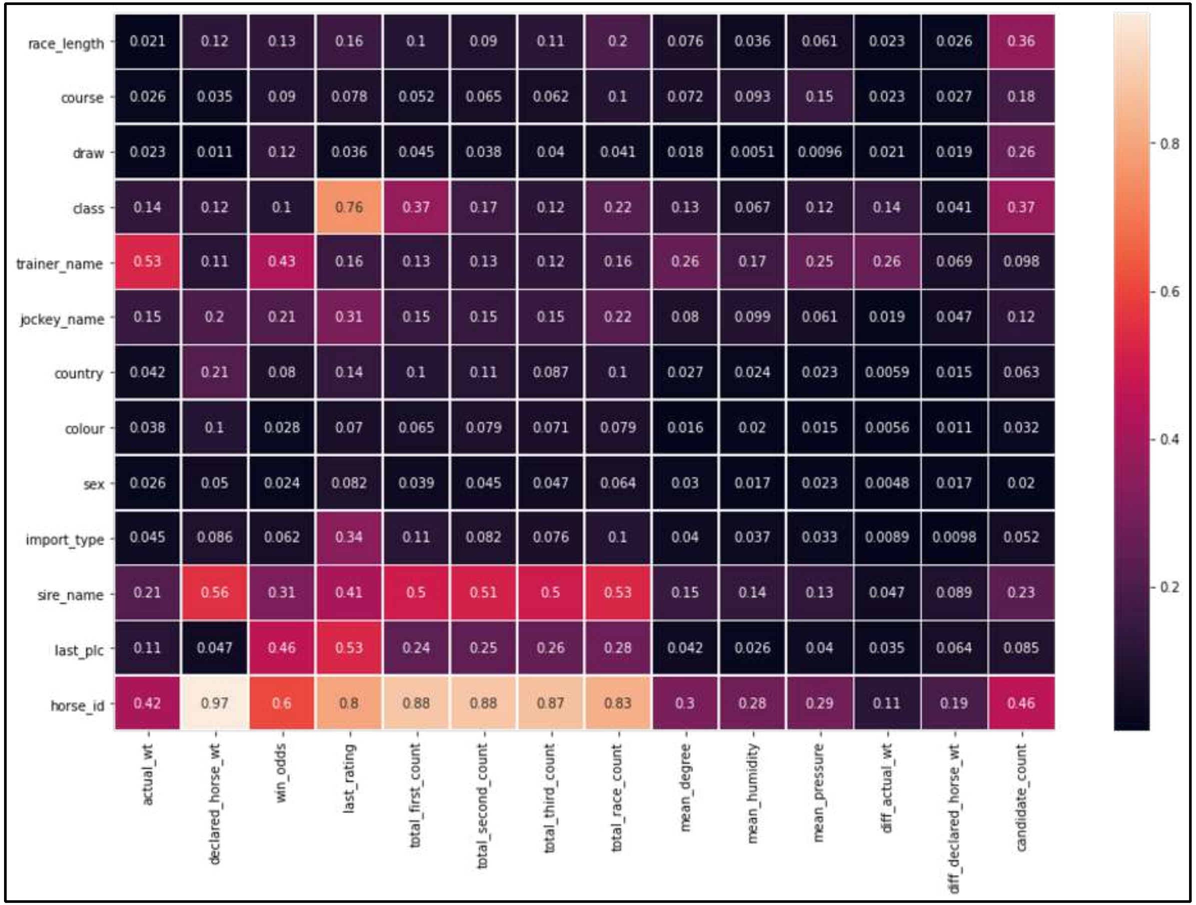
Data (Analysis)

- Association between categorical data
- Show whether the data is valid

Data (Analysis)

- Correlation ratio between continuous data and categorical data

- Show whether the data is valid



Data (Preprocess)

- Continuous data: z-score normaliztion

$$z = \frac{x - \mu}{\sigma}$$

- Categorical data: One hot encoding (YES/NO $\rightarrow$ {1,0})

XGBoost (Regressor)

- Developed by Tianqi Chen in 2014
- A scalable end-to-end tree boosting system
- As a regressor to predict the finishing time of the horses
- Result will be used in reinforcement learning and compared to the result of reinforcement learning

XGBoost (Reason using it)

- A lot of people have used it to win machine learning challenges
- Proved as a high efficiency and high accuracy system
- Easy to use and tune for different purposes

XGBoost (Process)

- Seperate the dataset to training set and testing set
- Training set is the dataset from 2014 to 2018
- Testing set is the dataset from 2019

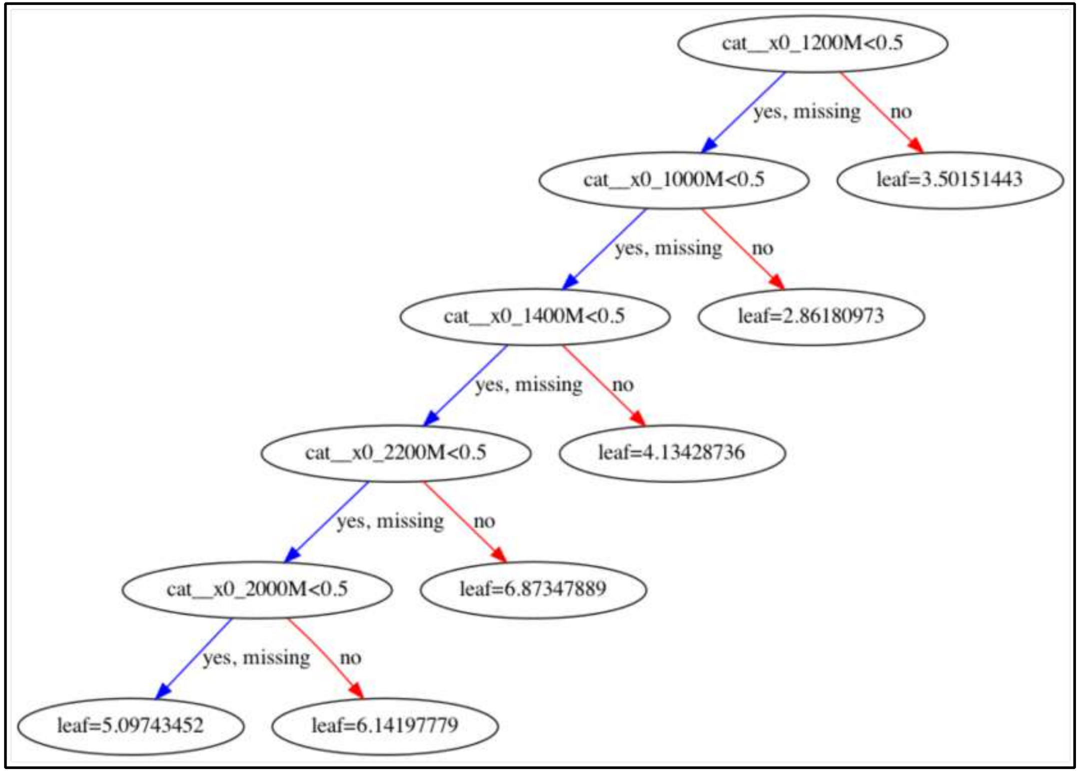
XGBoost (Hyperparameter)

```
XGBRegressor(base_score=0.5, booster='gbtree', colsample_bylevel=1,
             colsample_bynode=1, colsample_bytree=1, gamma=0, gpu_id=-1,
             importance_type='gain', interaction_constraints='',
             learning_rate=0.05, max_delta_step=0, max_depth=5,
             min_child_weight=1, missing=nan, monotone_constraints='()',
             n_estimators=230, n_jobs=0, num_parallel_tree=1, random_state=42,
             reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
             tree_method='exact', validate_parameters=1, verbosity=None)
```

XGBoost (Result)

- The tree generated by XGBoost

- Prove the model is successful



XGBoost (Result)

- R2 Score: 0.9974
- Accuracy of predicting the first place: 30.37%
- Accuracy of predicting the first, second, third place: 7.16%

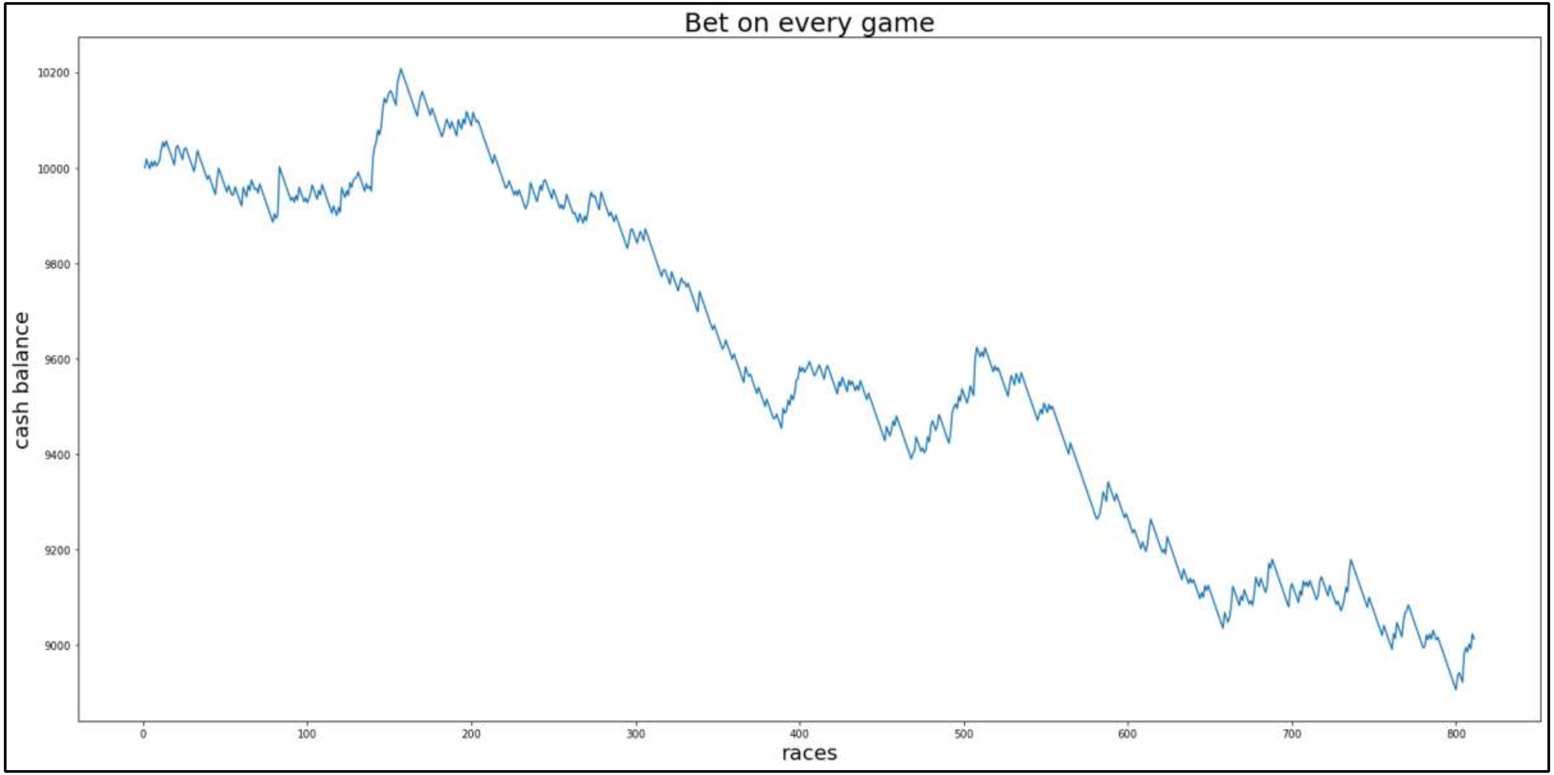
	0	df_date	race_date	season	race_index	horse_id	race_no	class	win_odds_	total_race_count_	finish_time_sec	rating	plc
9856	99.330414	2019-12-29	2019/12/29	19/20	293	HK_2018_C197	10	Class 3	3.9	12	100.15	64.0	3
9849	99.410896	2019-12-29	2019/12/29	19/20	293	HK_2018_C443	10	Class 3	5.1	4	100.69	72.0	8
9851	99.426056	2019-12-29	2019/12/29	19/20	293	HK_2017_B023	10	Class 3	5.8	25	100.83	63.0	11
9845	99.607193	2019-12-29	2019/12/29	19/20	293	HK_2017_B189	10	Class 3	8.9	35	100.49	77.0	5
9858	99.632103	2019-12-29	2019/12/29	19/20	293	HK_2014_T098	10	Class 3	12.0	69	100.13	61.0	2
9854	99.680313	2019-12-29	2019/12/29	19/20	293	HK_2017_B161	10	Class 3	10.0	17	100.30	72.0	4
9855	99.829094	2019-12-29	2019/12/29	19/20	293	HK_2016_A193	10	Class 3	13.0	45	100.82	63.0	10
9852	99.874359	2019-12-29	2019/12/29	19/20	293	HK_2017_B317	10	Class 3	16.0	21	101.25	70.0	12
9847	99.891357	2019-12-29	2019/12/29	19/20	293	HK_2017_B353	10	Class 3	15.0	18	100.51	67.0	6
9857	100.045280	2019-12-29	2019/12/29	19/20	293	HK_2017_B203	10	Class 3	25.0	25	100.09	70.0	1
9848	100.065971	2019-12-29	2019/12/29	19/20	293	HK_2015_V338	10	Class 3	33.0	41	100.51	77.0	7
9850	100.530182	2019-12-29	2019/12/29	19/20	293	HK_2016_A127	10	Class 3	69.0	19	102.16	76.0	14
9853	100.536362	2019-12-29	2019/12/29	19/20	293	HK_2017_B330	10	Class 3	74.0	14	101.71	78.0	13
9846	100.768089	2019-12-29	2019/12/29	19/20	293	HK_2018_C489	10	Class 3	102.0	3	100.69	69.0	9

XGBoost (Simulation)

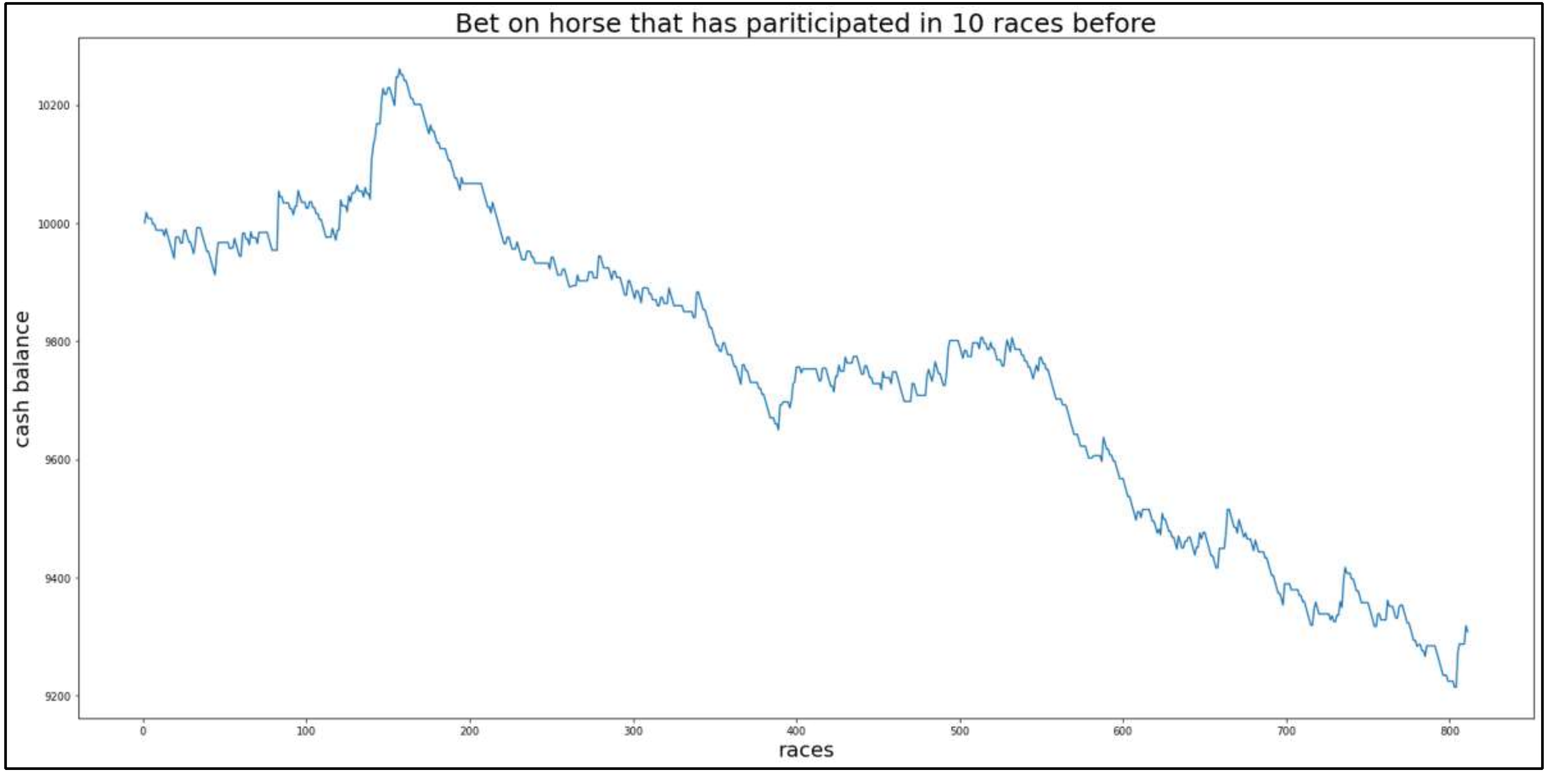
Betting Simulation:

- Each bet: $10
- Win: $10 * win_odds - $10 (Cost)
- Loss: -$10

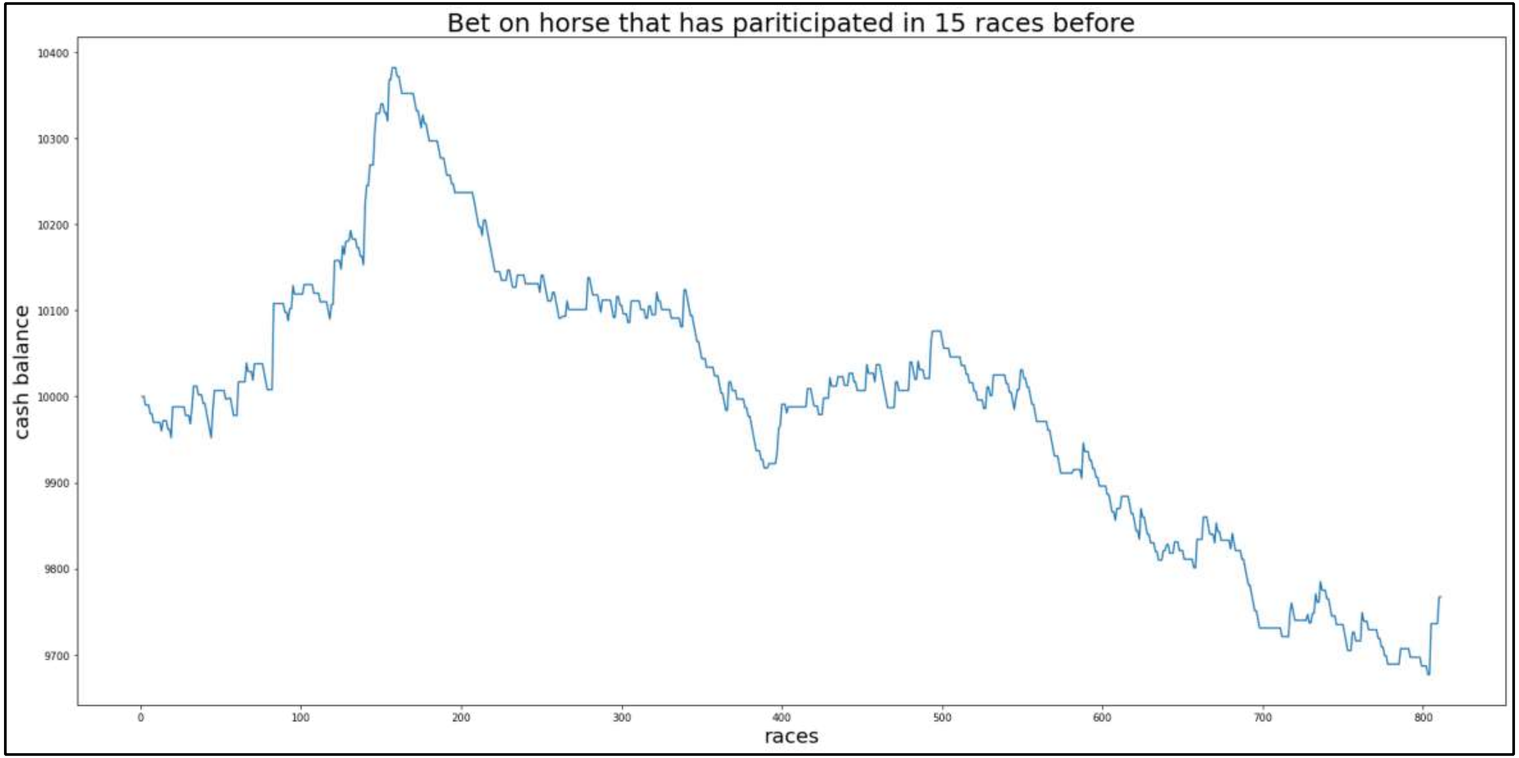
XGBoost (Simulation)



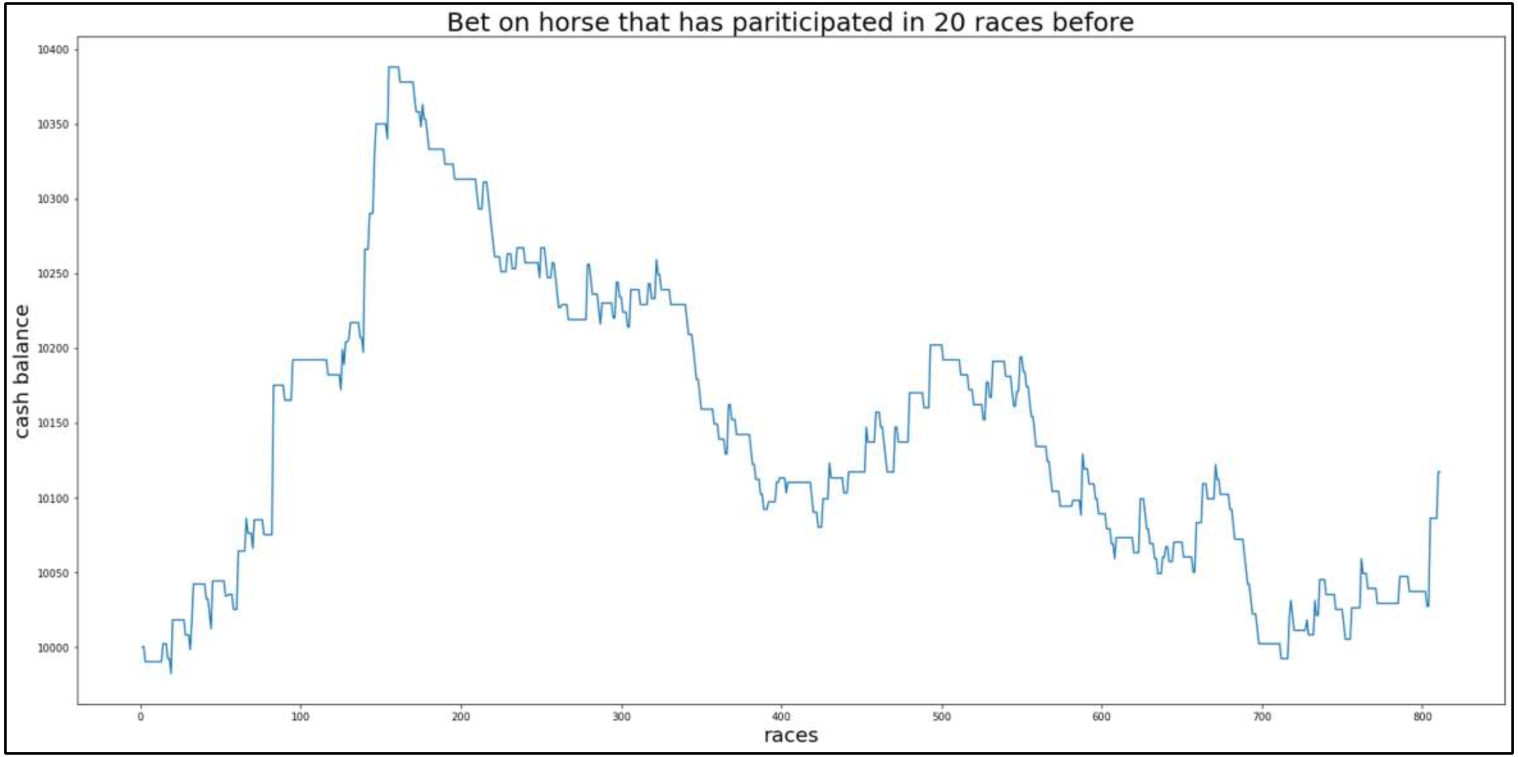
XGBoost (Simulation)



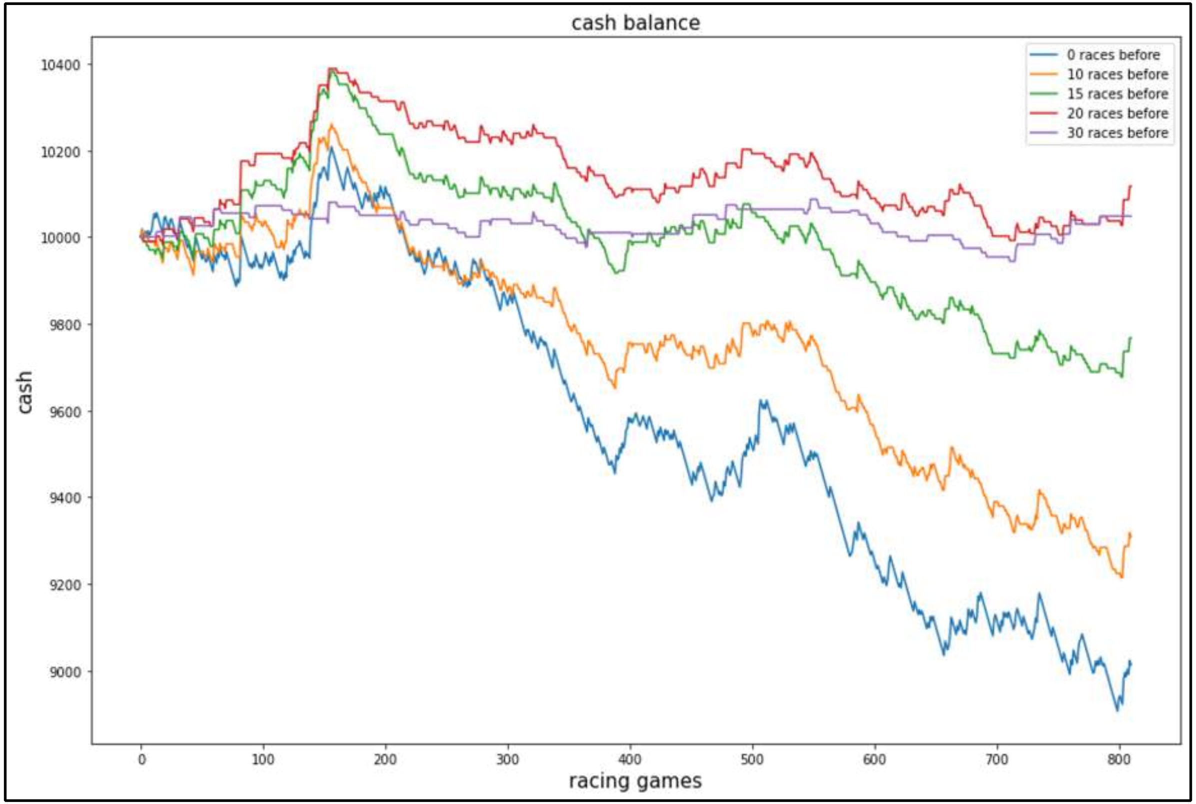
XGBoost (Simulation)



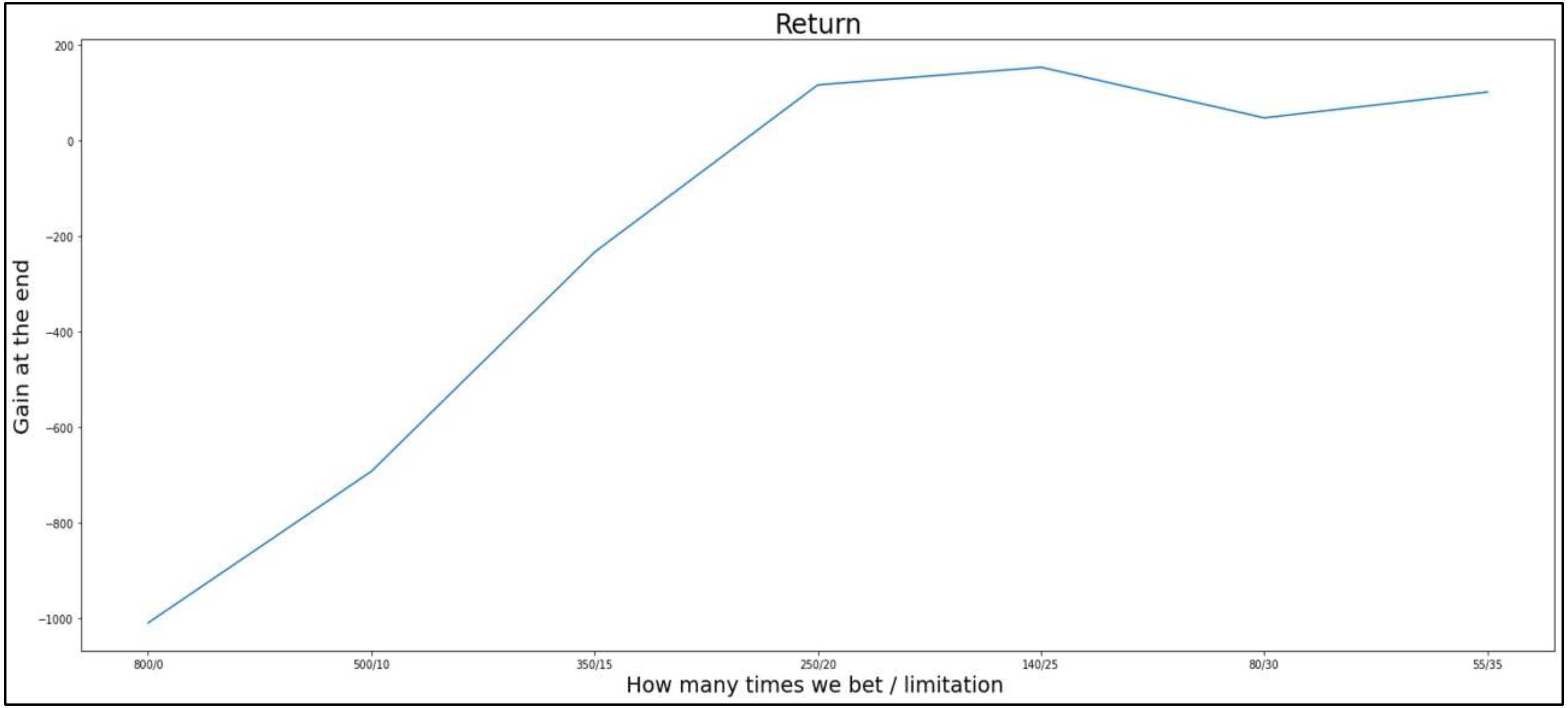
XGBoost (Simulation)



XGBoost (Simulation)



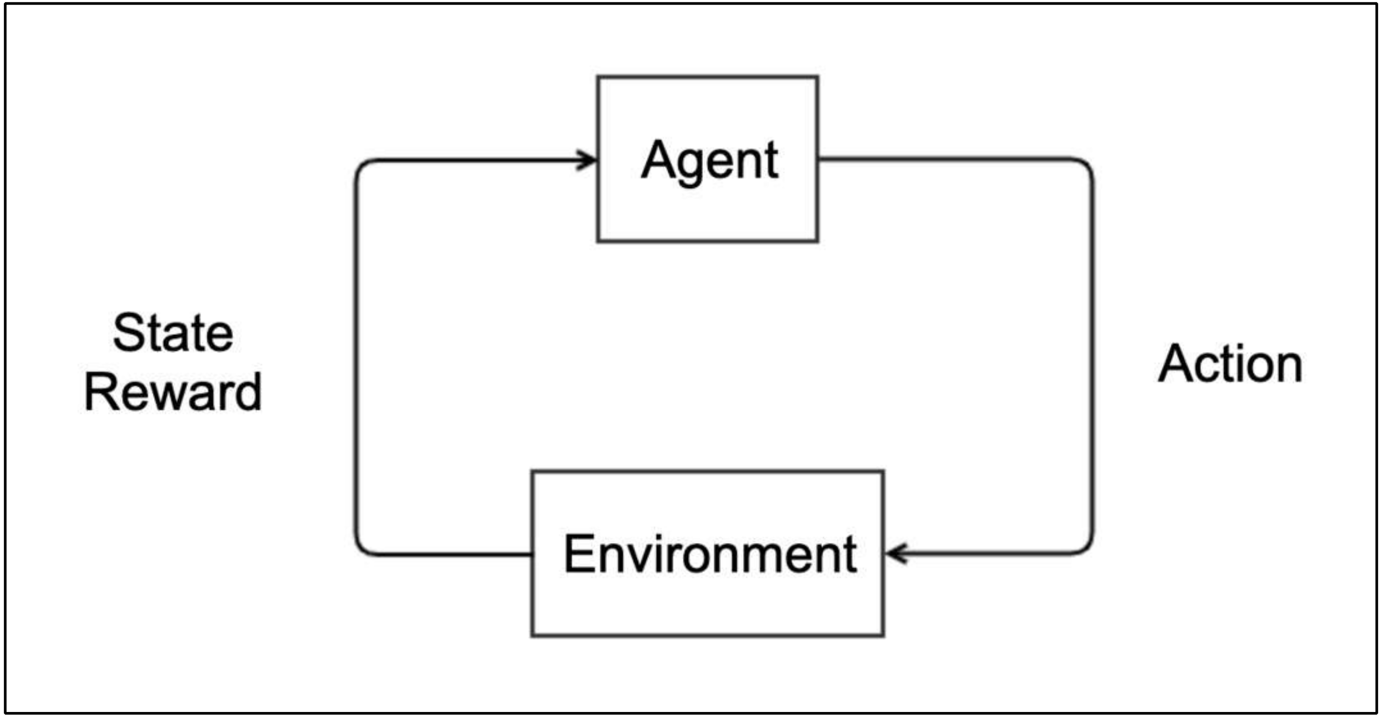
XGBoost (Simulation)



XGBoost (Conclusion)

- Positive correlation between participation experience and the win rate
- Although there are criteria, the return is not high
- Prediction with 30% accuracy can help in reinforcement learning
- Great experience of studying machine learning in horse racing

Reinforcement Learning (Algorithm)



Reinforcement Learning (Algorithm)

The goal of reinfocement learning is to find the best policy which can bring the best expected total reward: $E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$

Best policy: $$\theta^\star = \arg\max_\theta E_{\tau \sim p_\theta(\tau)} \left[\sum_t r(\mathbf{s}_t, \mathbf{a}_t) \right]$$

Reinforcement Learning (Objective)

Value-based algorithm:

- Improve the policy based on value (reward)
- Similar to the situation of gambling
  - We place a bet on a horse since we believe the horse will bring us money
  - The agent place a bet on a horse since he believes the horse will bring him reward

Deep Q-learning!!!

Reinforcement Learning (Q-learning)

- Value-based algorithm
- Q-funtion: Q(s,a)
- Q-table:

State \ Action	a1	a2
s1	value1	value2
s2	value3	value4

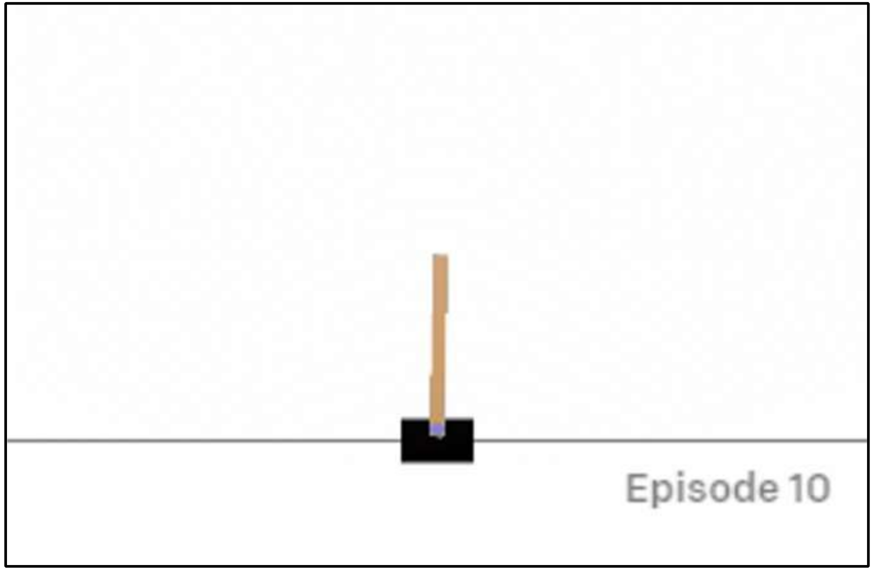
Reinforcement Learning (Q-learning)

Algorithm of Q-learning:

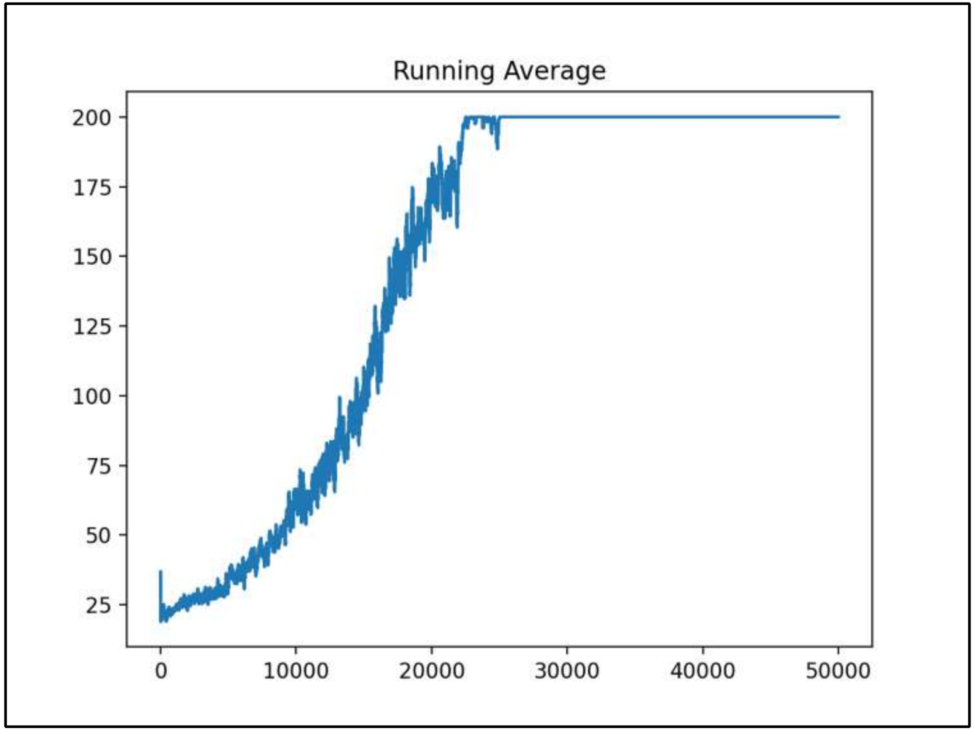
1. Initialize Q function Q(s,a) to some random values
2. Take an action from a state using epsilon-greedy policy from Q function
3. Observe the reward and the new state
4. Update the Q table by : $Q(s,a) = Q(s,a) + \alpha(r + \gamma max Q(s'a') - Q(s,a))$
5. Repeat step 2 to step 4 until terminal state

Reinforcement Learning on playing games

- “Cartpole” from OpenAI gym
- Prevent the pole falling over the cart



Reinforcement Learning on playing games



Reinforcement Learning on playing games

Reason not using Q-learning:

- Only can be used in some simple problems or games
- Lots of data causing a huge Q-table and the efficiency is too low to complete the Q-table
- Impossible to test since it will only take random action while meeting new races with new horses.

Reinforcement Learning on horse racing

- Deep Q Learning with MLP policy
  - Use neurall network to approximate Q-function
  - Loss Function: $E_{s,a,r,s'}[((r+\gamma \max_{a'} Q(s',a';\theta')) - Q(s,a;\theta))^2]$
    - Stochastic gradient descent

- Moving target problem
  - change θ -> θ’ will be affected
    - Freeze the θ’ long enough from DeepMind

Reinforcement Learning on horse racing

Environment

- Observation Space
  - the features of 14 horses, including invalid horses (set to -99)
- Action Space
  - 15 actions : {‘bet on horse 1’, ... , ‘bet on horse 14’, ‘do not bet’}, referring to the input order
  - bet with a fixed amount 10 dollars
- State
  - 1 state = 1 horse racing game
- Termination state
  - Lose more than $1000
  - goes through all the horse racing games

Reinforcement Learning on horse racing

Reward Function

- Idea
  - Reward of winning $100 > Reward of winning $11
  - $$R(\text{bet and win}) = C_1 * \Delta \text{Cash Balance}, \text{ where } C_1 > 0$$
    $$R(\text{bet but lose}) = C_2 * \Delta \text{Cash Balance}, \text{ where } C_2 < 0$$
    $$R(\text{ do not bet }) = C_3 * \text{win odds of the true first place}, \text{ where } C_2 < C_3 < 0$$

Reinforcement Learning on horse racing

Invalid action

- Ignore it
  - not reasonable, cheaper version of ‘do not bet’
- Same penalty as ‘do not bet’
  - it will treat betting on invalid horse as ‘do not bet’ if there are less than 14 horses
- Large penalty
  - Bet only on races with 5 horses OR ‘do not bet’

R(bet on invalid horse) = R(do not bet) is chose

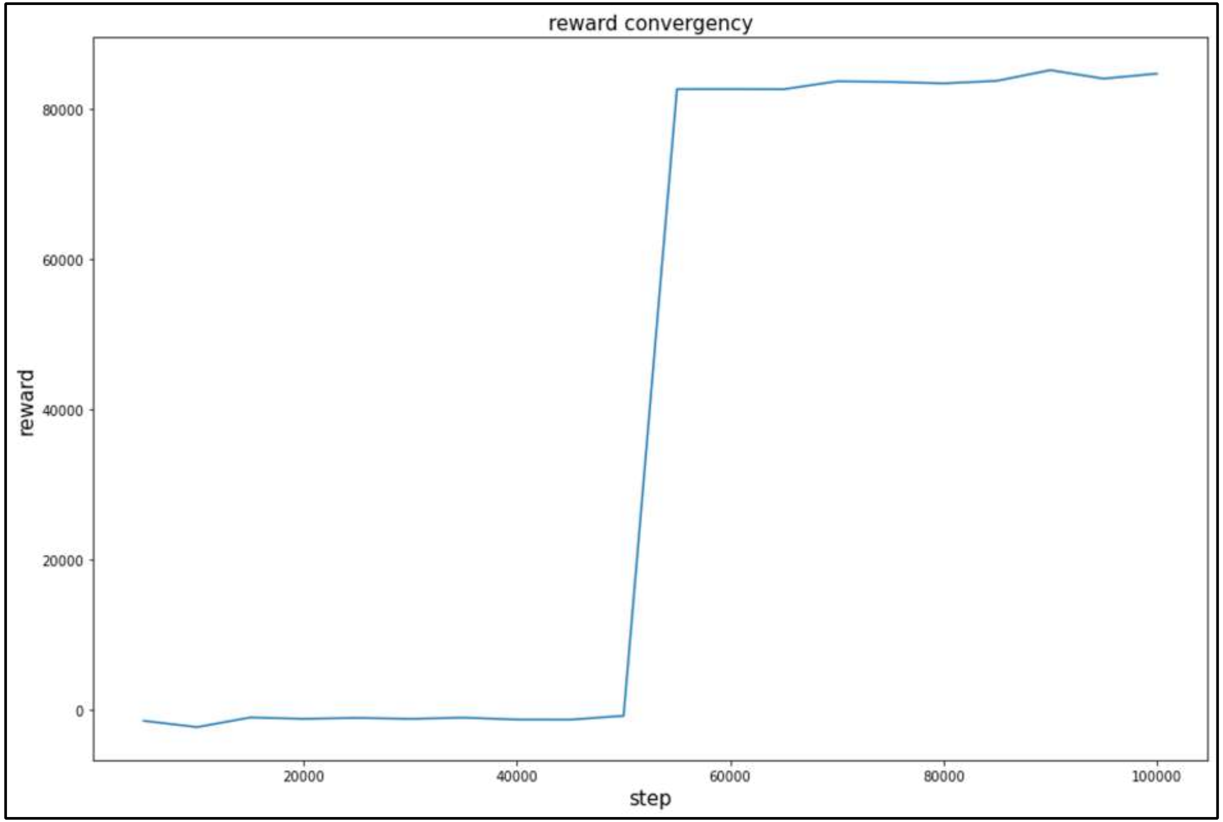
Reinforcement Learning on horse racing

Input Order

- Shuffle the order of the horse -> do not converge on reward

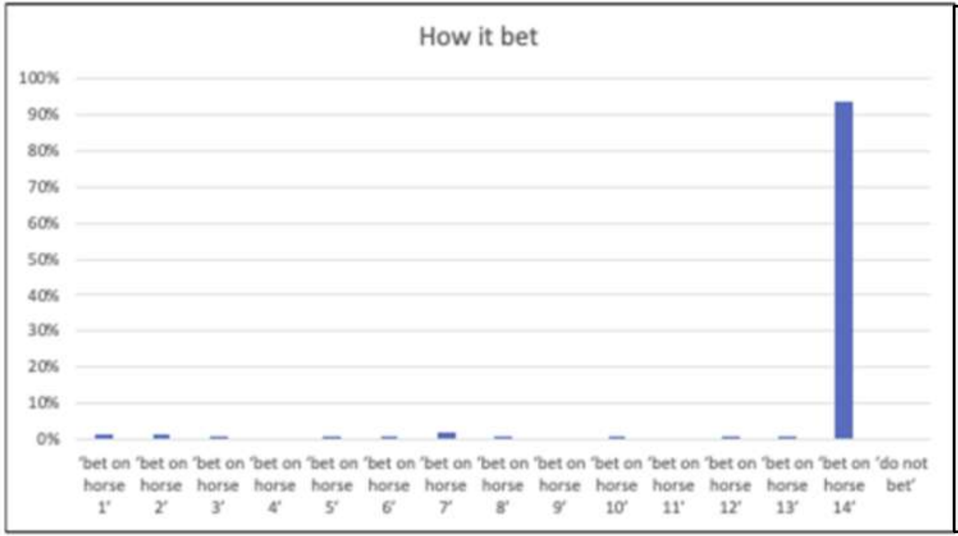
- We use the prediction from XGBoost to order the input
  - The horse with largest number is the fastest horse we predicted in a race
  - If there are only 13 horses, then the horse 14 is a invalid horse
    - to see if the agent learns the exist of 'invalid horse'

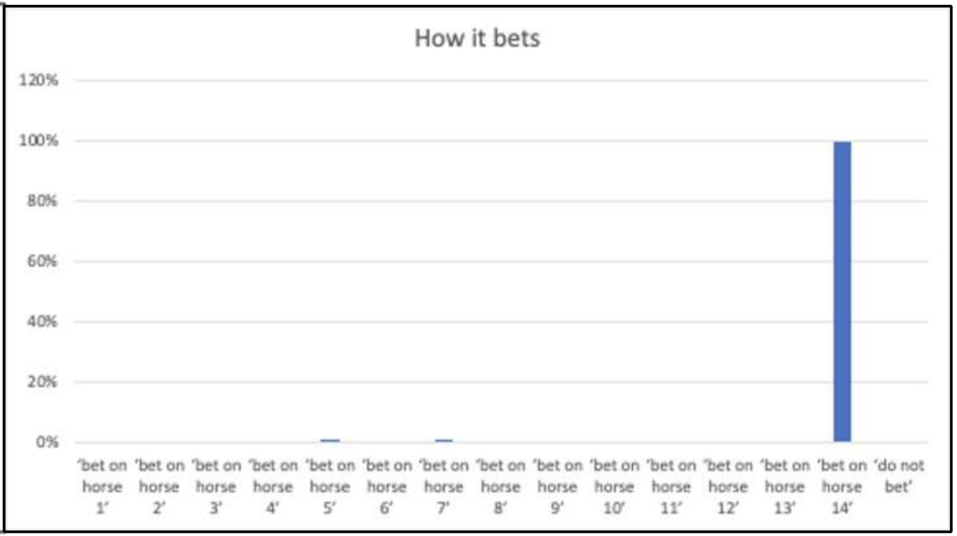
Convergency



How agent bet

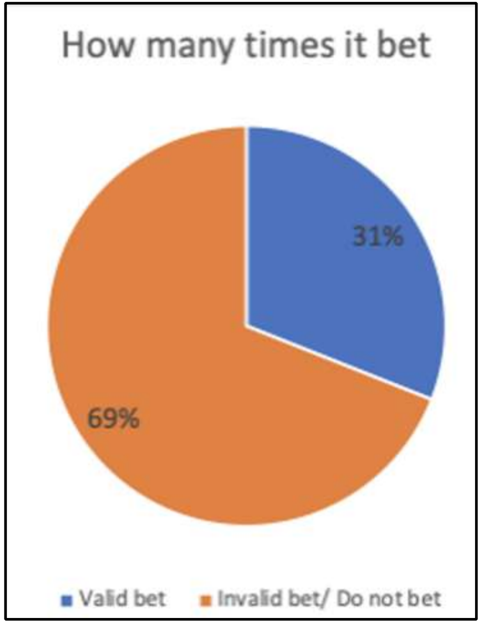
training set



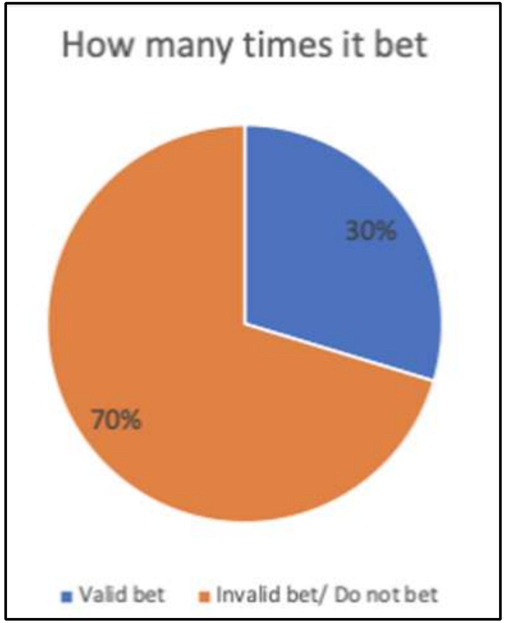


Invalid and valid betting

training set

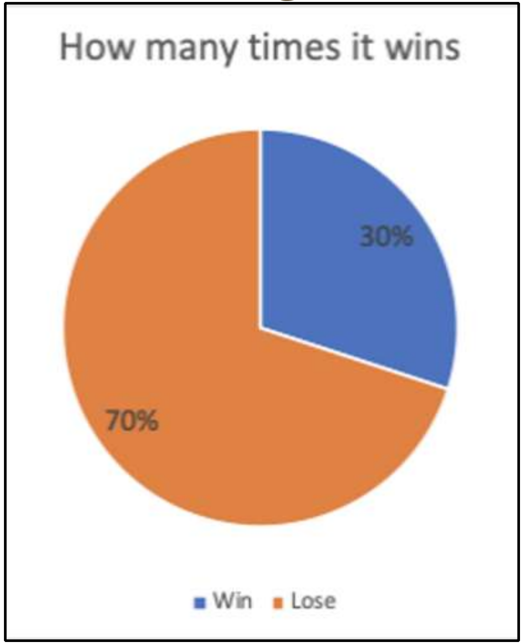


testing set

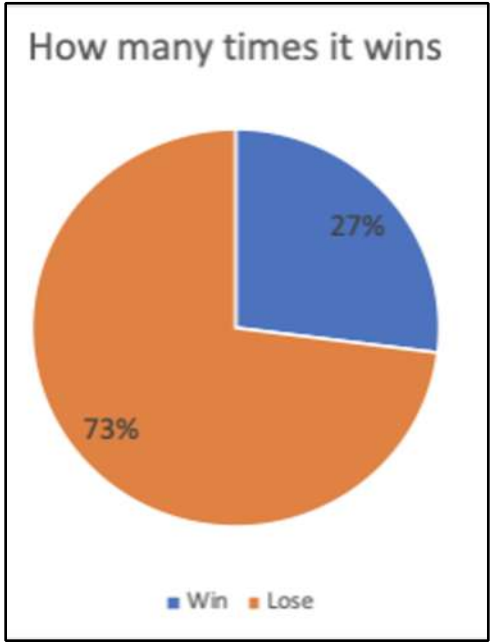


Win ratio

training set

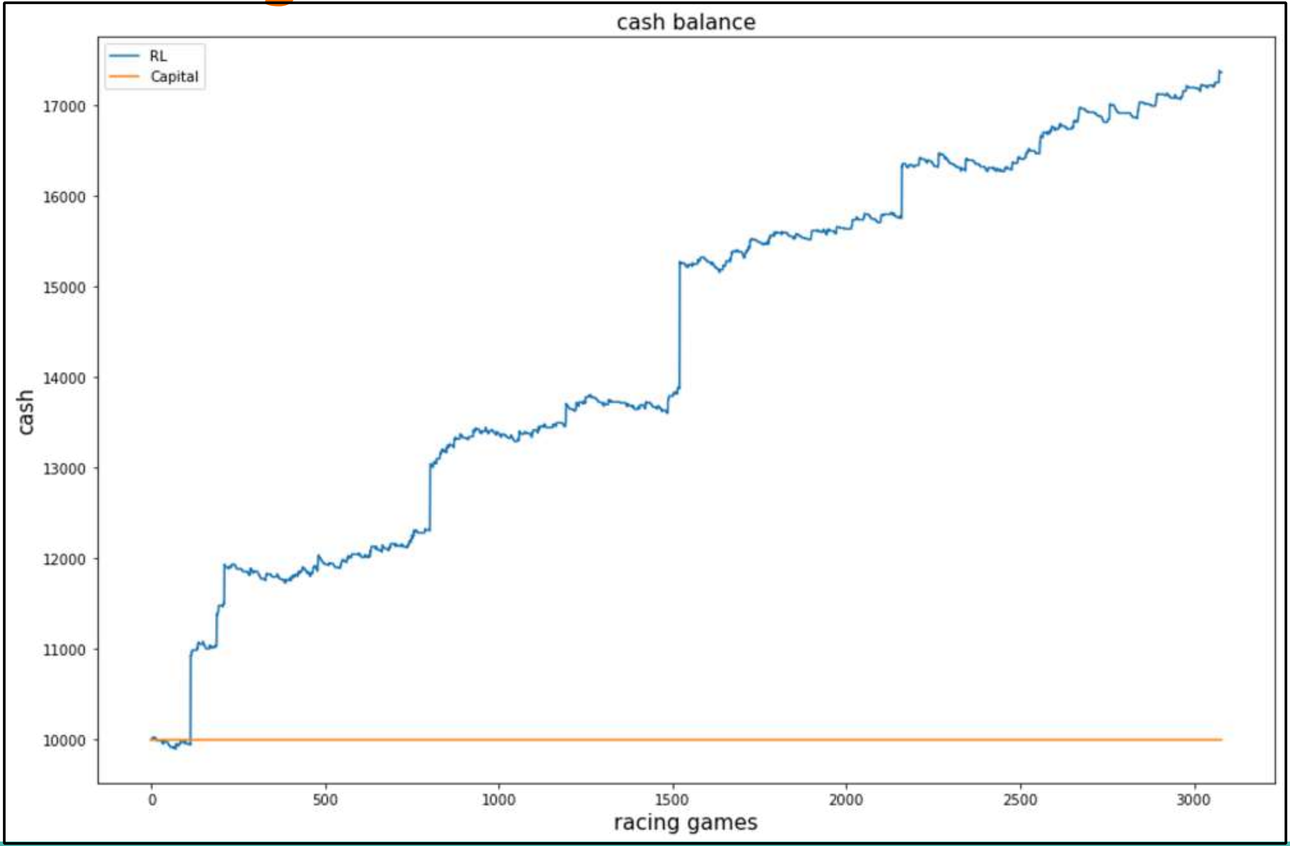


testing set



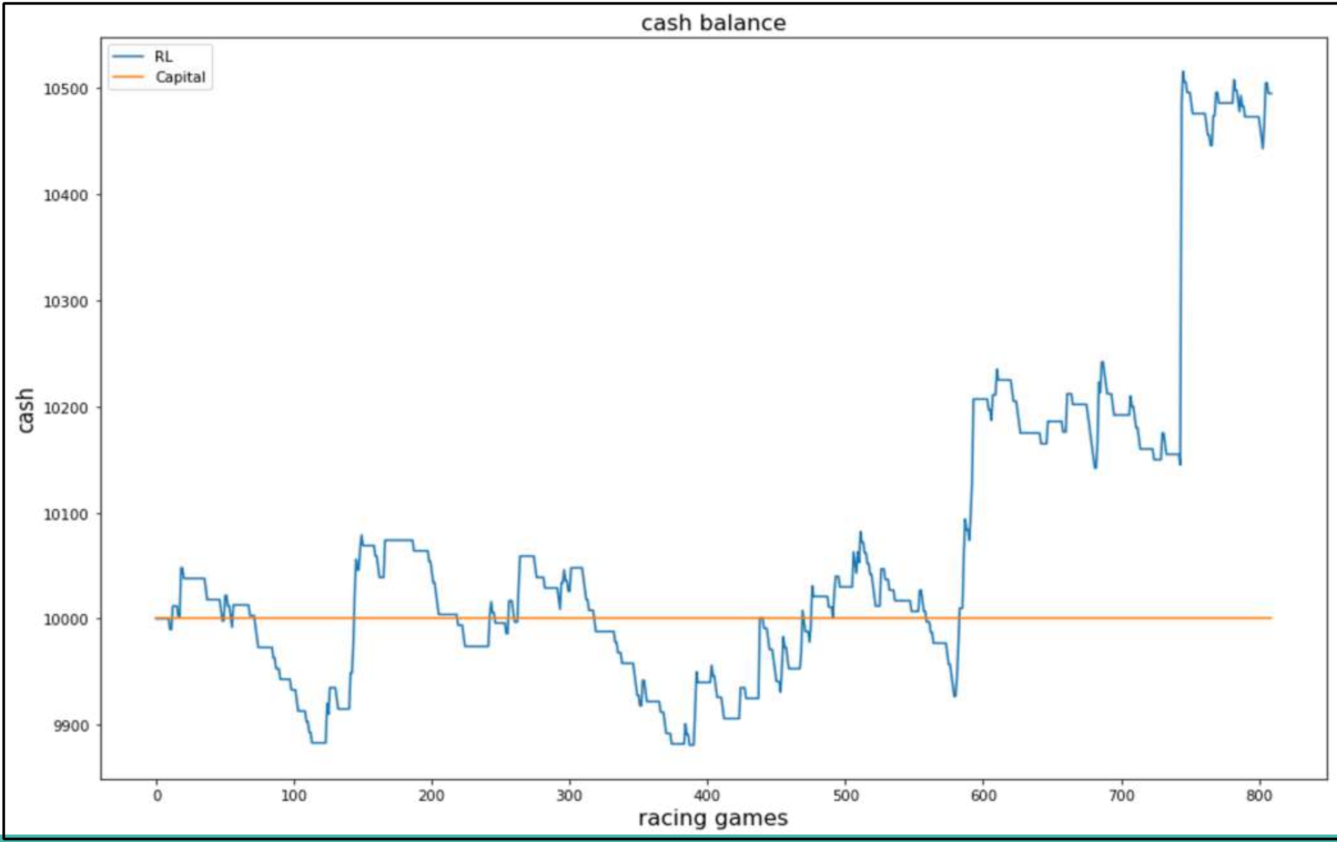
Cash balance in training set

Largest win : 10 * 140



Cash balance in testing set

Largest win : 10 * 34



Result analysis

- 0 ‘do not bet’ action
  - Same as our expectation,
    - The agent treat ‘bet on invalid horse ‘as ‘do not bet’
    - It means the agent bets on all the races with 14 horses
      - can’t learn the meaning of the aciont ‘do not bet’?
      - betting on every races with 14 horses is a nice choice?
- Large win odds
  - Bet on those horse with high win odds to gain a larger reward

Result analysis

- Bet on races with 14 horses ONLY
  - Reward is maximized when it only bet on the races with 14 horses
  - Penalty of losing is larger than betting on a invalid horse
  - More explanation in the next result

- ‘Bet on horse 14’ most of the time
  - The horse 14 is the most likely the winning horse in races with 14 horses
  - This is the most safe action
    - 30% accuracy in races with 14 horses
    - ‘invalid horse’ most of the time, which is do not bet
      - the penalty is less than ‘losing’

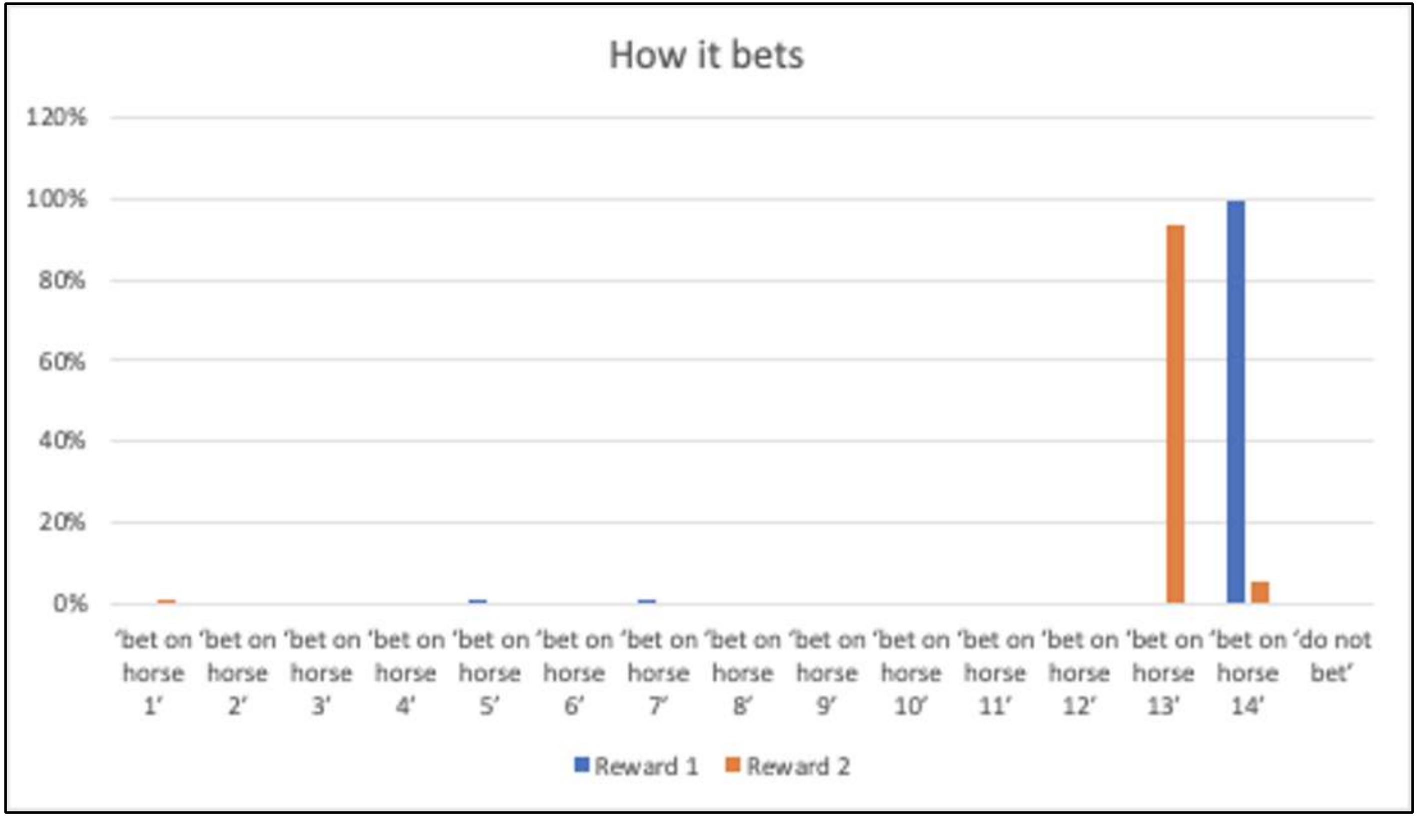
Improve the reward function

We want to encourage the agent to bet more, not just limited to races with 14 horses.

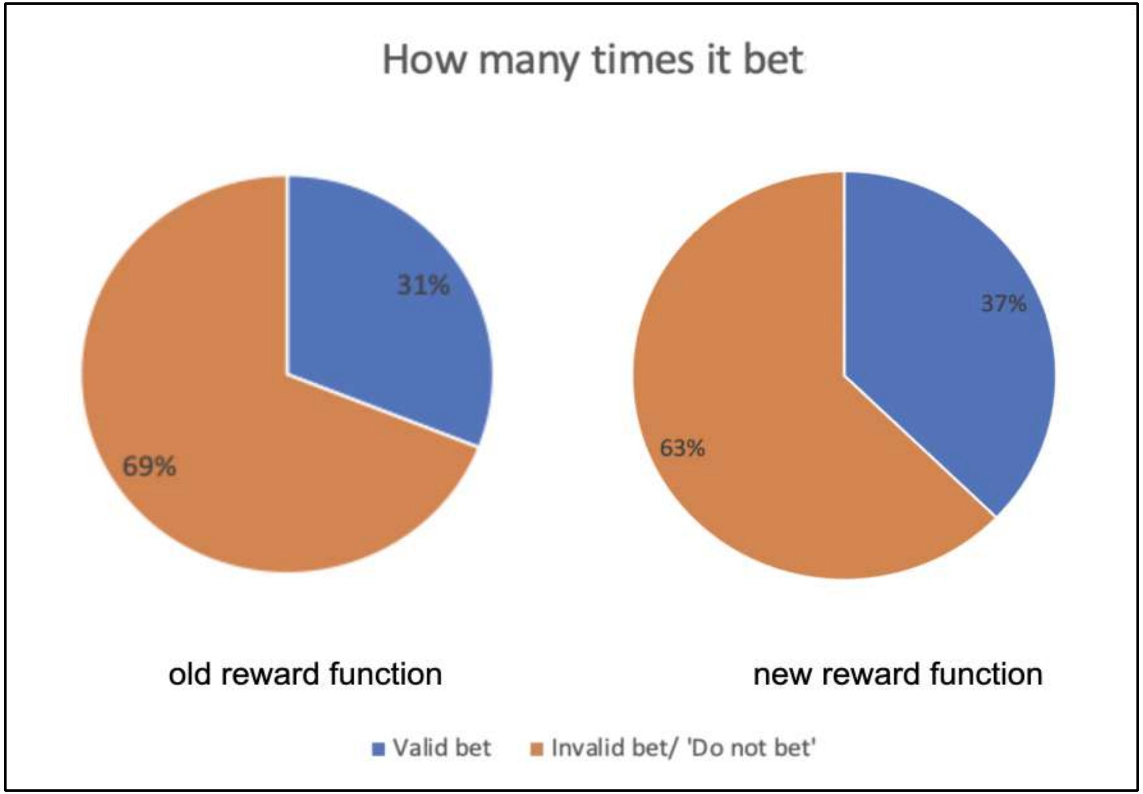
Solution:

- Increase the reward of ‘winning’
- Decrease the penalty of ‘losing’
- Decrease the penalty of ‘invalid betting’

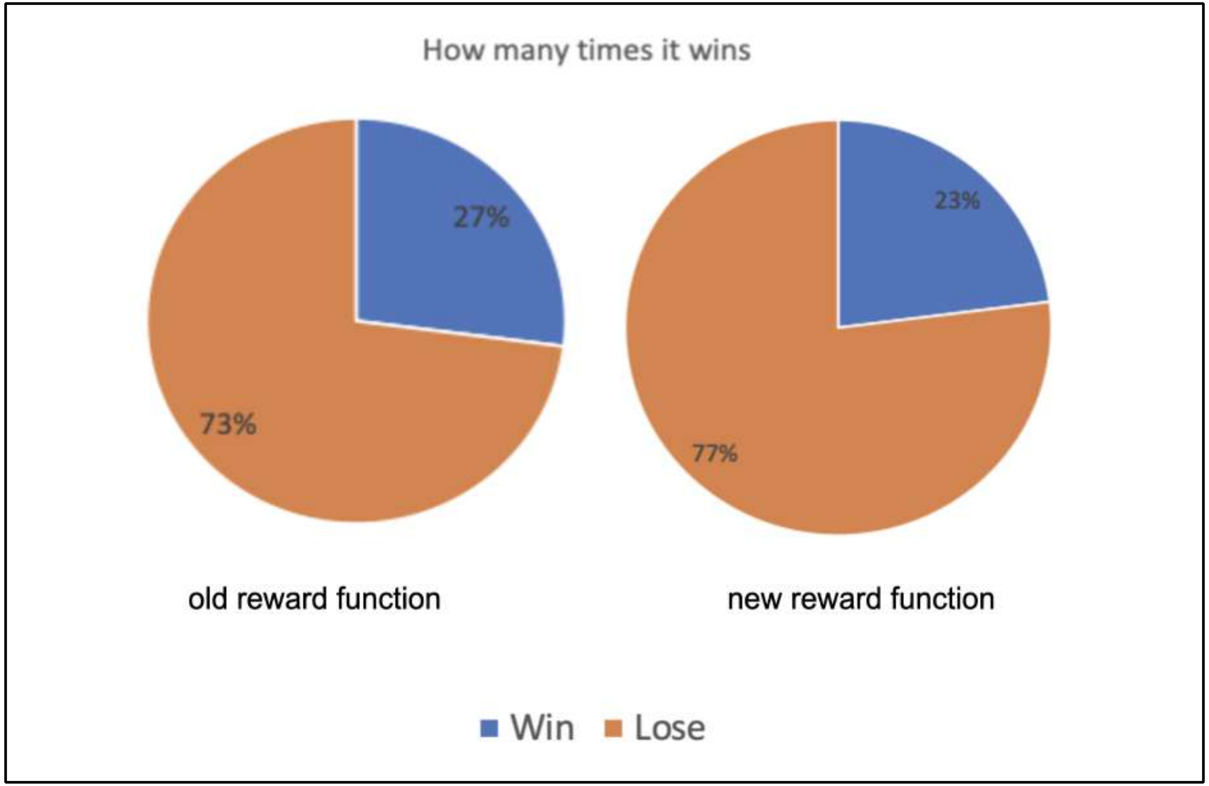
Comparison of how agent bet



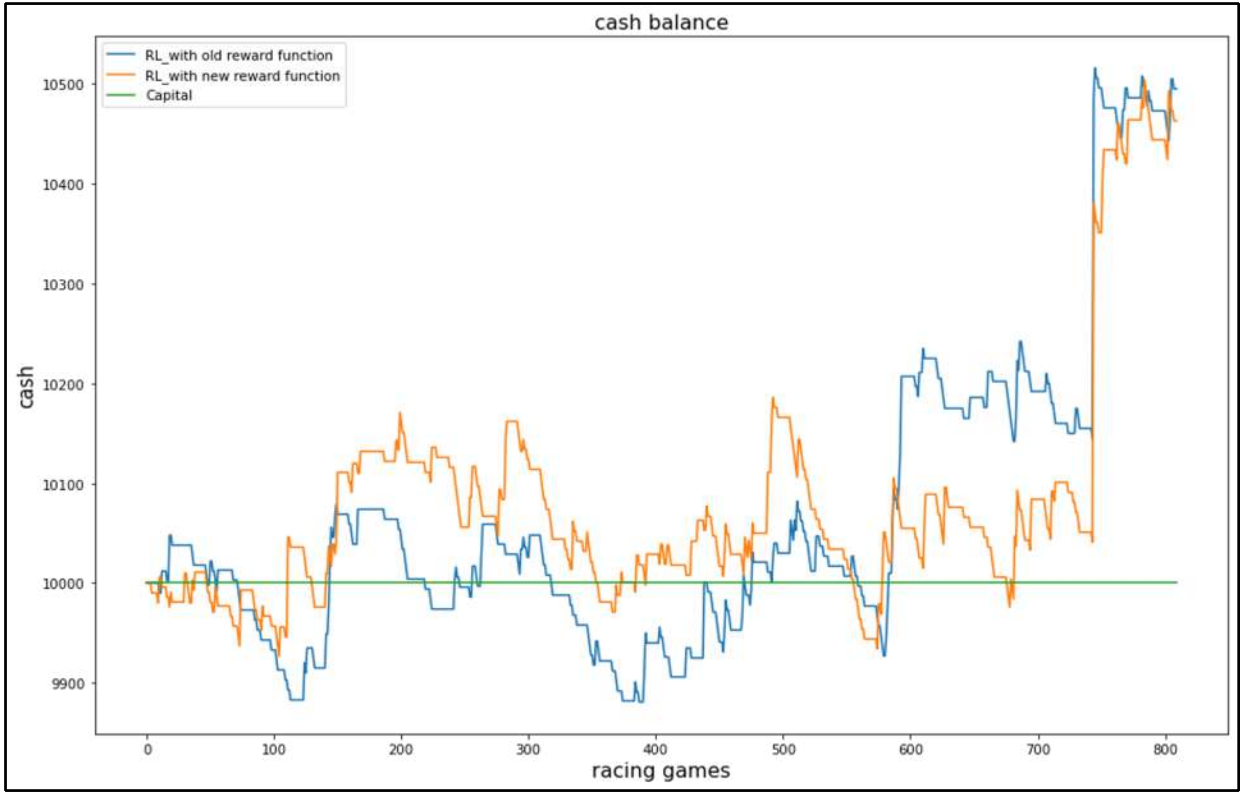
Comparison of Invalid or valid betting



Comparison of Win ratio



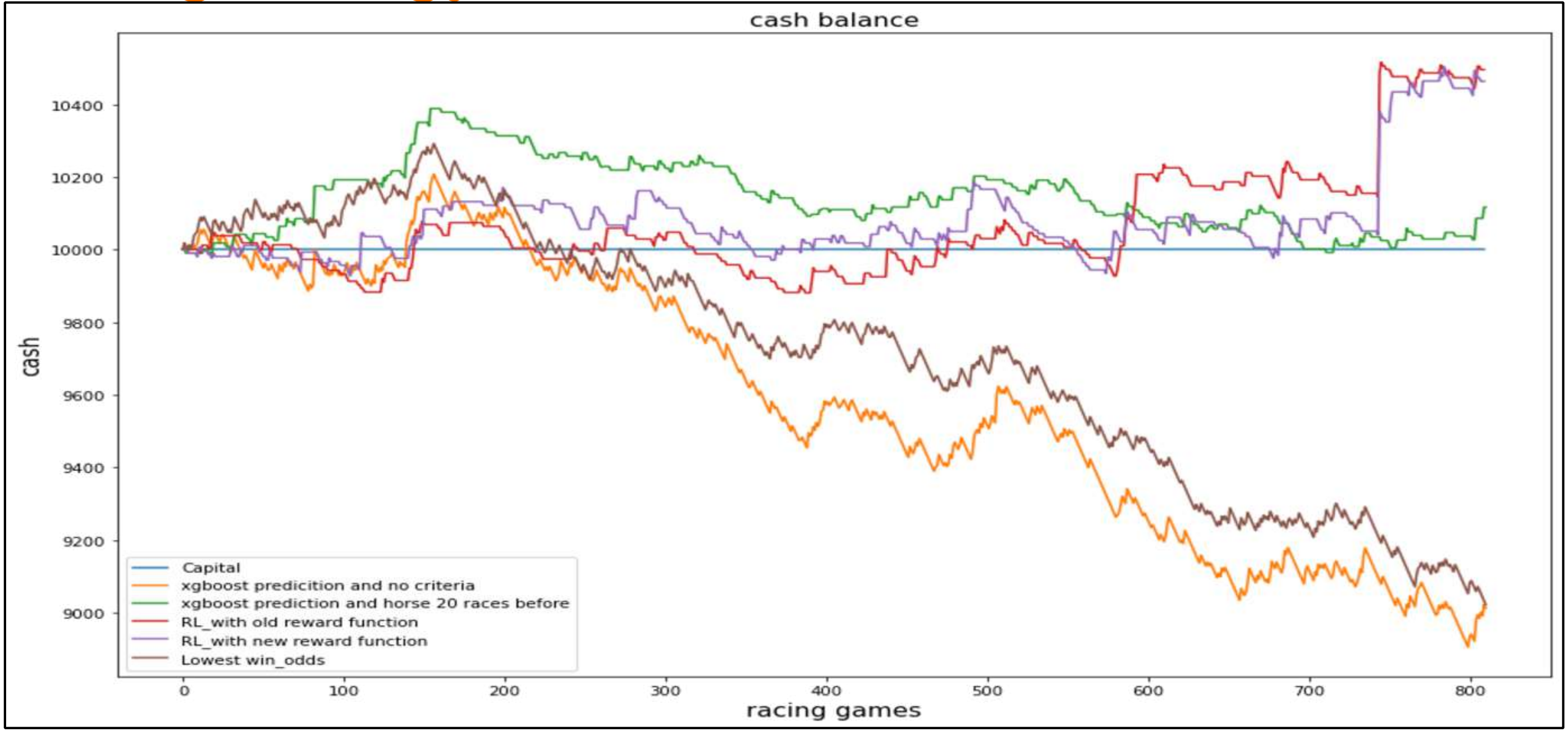
Comparison of cash balance



Result analysis

- 0 ‘do not bet’ action
  - Same as our expectation,
    - The agent treat betting on invalid horse as ‘do not bet’
    - It still bet on all the races with 13 and 14 horses
      - Best choice?
      - cant learn well?
- Win ratio is decreased
  - It is the reason why the old agent refused to bet on races less than 14 horses
    - It will lower the win rate and the reward is less than before
- ‘Bet on horse 13’ most of the time
  - Refuse to bet more
    - The win ratio is decreased
    - The reward is maximized
  - So It is still limited to bet only on races with 13 and 14 horses

Betting strategy



Conclusion

- Input order/format matters!
  - Invalid horse
  - ordered by the win odds?
- Bet on races with particular number of candidates only
  - maybe build up a specified model
    - 14 horses RL model
    - 13 horses RL model …
- Combinining all the races
  - races with different counts of candidates may become the ‘noise’ to each other
- Construction of reward function
  - It is hard to balance the reward and penalty
  - It will affect how the agent bet

Conclusion

- More Betting types
  - Quinella
  - Place
  - …
- Different betting amount
  - The betting amount should not be fixed
- Based on the needs above
  - different model is required.
    - policy gradient or actor-critic
    - continuous output

Thank You!

Q&A