Bandit Algorithm, Reinforcement Learning, and Horse Racing Result Prediction

LYU2103
Wong Tin Wang David(1155127053)
Sze Muk Hei(1155127477)
Supervised by Prof. Michael R. Lyu

Objectives

- Predict the horse racing result accurately comparing to the previous FYP
- Generate stable profits using reinforcement learning techniques at the end of the whole project

Agenda

1 Introduction

2 Data

3 Horse Racing Prediction

4 Horse Betting Using MAB

5 Conclusion

6 Q&A

1

Introduction

Backgrounds, Motivations & Objectives

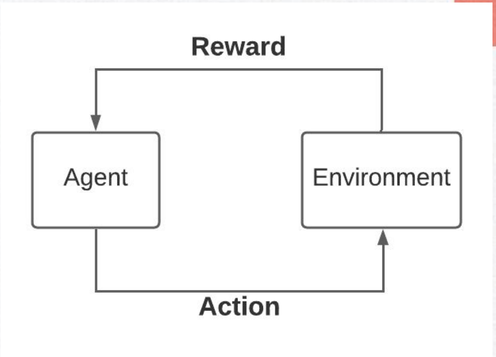
Backgrounds

- Horse racing in Hong Kong started since **1846**
- **Horse racing prediction has been widely studied**
    - Deep Learning
    - XGBoost
    - SVM-Based committee machine
- **Betting Rules:**
    - WIN (1st place)
    - PLACE (any 1 of the top 3 horses)

Backgrounds

- **Reinforcement learning**
    - Agent and environment

- **Multi-armed Bandit (MAB)**
    - One of reinforcement learning algorithms
    - Explore-exploit dilemma



Motivations

- Horse racing in Hong Kong is a popular betting events since last century
- Generate profits
- Reinforcement learning is rarely considered for horse betting
  - Especially for MAB
- Data transparency

WHAT?
WHY?
HOW?

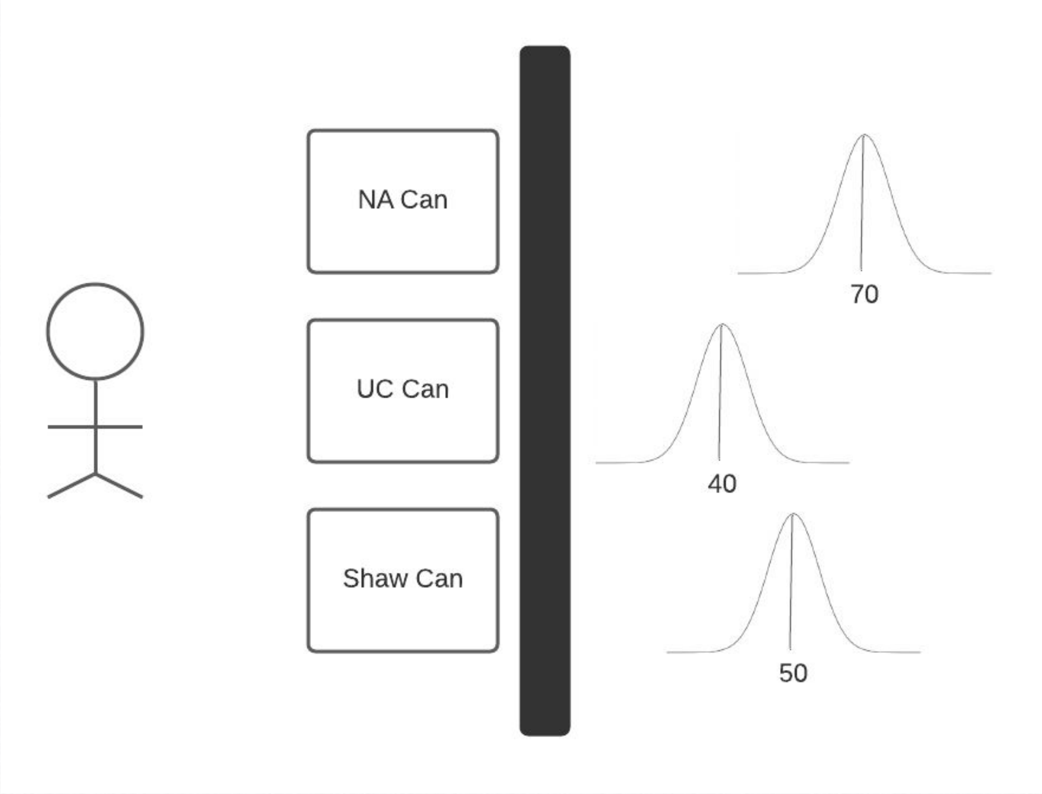
Explore-Exploit Dilemma

Advertisements Selection

Article Recommendation

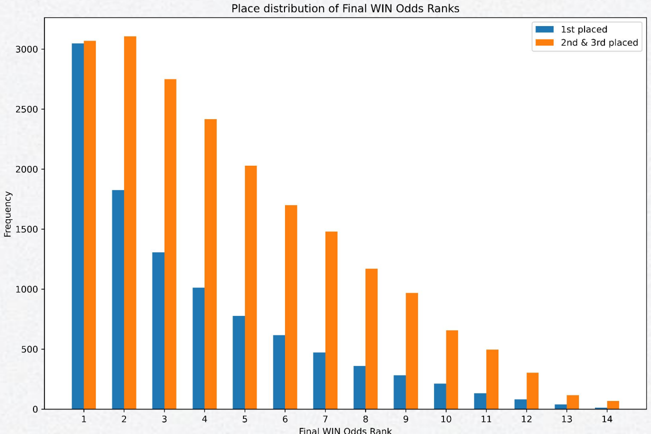
Portfolio Allocation

Explore-Exploit Dilemma



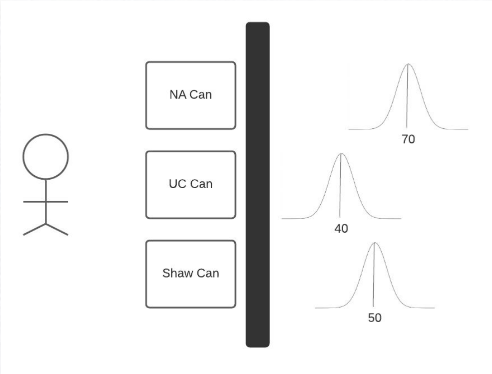
Why we use MAB?

- **Explore-Exploit dilemma in horse betting**
  - Bet on the horse most likely to win but with lower odds
  - Bet on the horses with higher odds that less likely to win
- **Maximize the profits**
- **No one used MAB in horse betting as far as we know**



MAB Algorithms

- **Epsilon-Greedy Method (constant exploration)**
  - Define an Epsilon (5%, 10%)
  - Use for limit the frequency of exploration
  - Explore randomly during the process
- **Other algorithms (adaptive exploration)**
  - Upper-confidence-bound (UCB)
  - Thompson Sampling (TS)

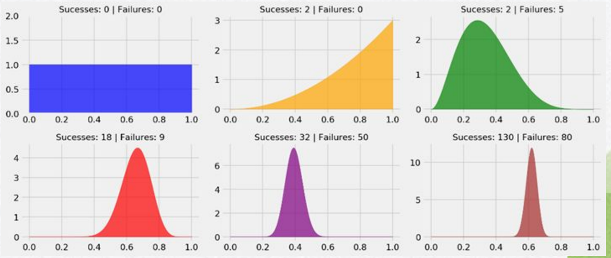


Upper-confidence-bound (UCB)

- Estimate reward of an action by sample mean û.
  û should differ from actual mean u by error Δ. Then u is within the confidence bound $\hat{u} - \Delta \leq u \leq \hat{u} + \Delta$.
- Δ is set in the way that it reduces as the action is played more. That means we have higher confidence that û would be close to u.
- UCB algorithms always choose action with highest upper confidence bound $\hat{u} + \Delta$. This is called "optimism in the face of uncertainty" as the reward could be as high as this bound.
- So the ucb is large at first, making all actions have high chance to be played (explore). And as we play more, only those with high actual mean will be played as Δ is small (exploit).

Thompson Sampling (TS)

- Assume reward follow certain probability distribution
  - Bernoulli
  - Gaussian
- Estimate parameters for the distribution and draw a random value as estimation for mean reward
- Bernoulli TS
  - Use Beta distribution as the prior to calculate p(q | reward)
  - Beta distribution changes depends on observed reward
  - Beta would concentrate around actual q



Terminology

- **Actions Set**
  - Include 3 actions (canteens)
  - Trying a new canteen (Explore)
  - Going back to the best canteen tried before (Exploit)
- **Optimal reward**
  - In this case: 150 x 70 = 10500
- **Regret**
  - Difference between actual rewards and optimal
  - As small as possible
  - For explore only: 10500 - (50 x 40 + 50 x 50 + 50 x 70) = 2500

How we use MAB?

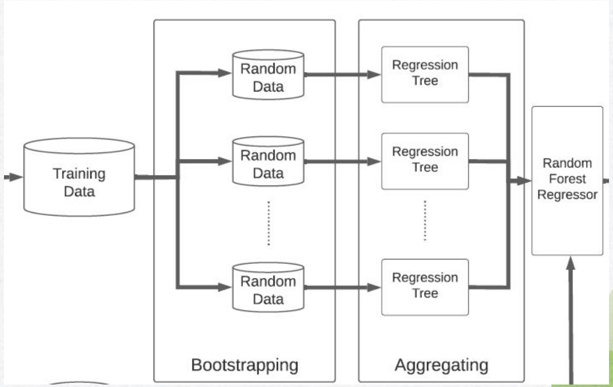
- **Not apply to horse racing prediction directly**
  - Unable to define all horses as action set
    - Some horses participated in race once only
    - Difficult to estimate their performance
  - Insufficient data
- **Our approach:**
  - Use **Random Forest** as horse racing prediction
  - Use **MAB** for betting only with the result prediction

Why we use Random Forest?

- **High Interpretability**
  - Determine the significant factors in prediction
- **Avoid overfitting or underfitting**
  - Tree ensembling (bagging)
- **Reduce data pre-processing work**
  - No need to normalize the data
- **Able to handle complex datasets**
  - High dimensionality
- **Stable result for MAB training**

How Random Forest Works?

- **Impurity function**
  - Mean squared error
- **Trees bagging**
  - Bootstrapping datasets
  - Aggregating results
- **Random sampling**
  - Randomly pick features for node splitting



Contribution

1. Created a combined approach
    - Random forest
        i. WIN accuracy: 24.537%
        ii. PLACE accuracy: 47.153%
    - MAB
2. Figured out factors with
    - Strong correlation with the outcome
    - Crucial to our prediction
3. Explored multi-armed bandit algorithms
    - Horse betting
    - Feasibility of generating profit

2

Data

Descriptions, Analysis & Pre-processing

Sources & Descriptions

- **Data Sources**
    a. The Hong Kong Jockey Club
    b. Data Guru
    c. hkHorse
- **Datasets**
    - Ranged from 1979 to 2021
    - Tables:
        - Races data
        - Horses data
        - Horse-race data
        - Betting odds data

Races Data

Features	Description	Types	Samples
raceid	A unique id of a race constructed with race date, distance, track type and race season id	Categorical	1979-09-22-001-1235-Grass
raceidseason	Index of a race in race season	Index	001
racedate	Date of the race	Index	1979-09-22
racetrack	Racecourse of the race	Categorical	HV, ST
tracktype	Type of racetrack of the race	Categorical	Turf, Grass, Sand, AWT
course	Width of inner rail and outer rail of the track	Categorical	(Show in later table)
distance	Distance of the race (in meters)	Categorical	1000, 1200, 1400, 1600, 1650, 1800, 2000, 2200, 2400
going	The condition of the track	Categorical	(Show in later table)
raceclass	The race class of horses in the race	Categorical	1, 2, 3, 4, 5

Horses Data

Features	Descriptions	Types	Samples
horseid	A unique id of the horse	Categorical	HK_2016_A061
horsename	A unique name of the horse	Categorical	RATTAN
country	The country of the horse	Categorical	NZ, AUS, IRE
colour	The colour of the horse	Categorical	Brown, Grey, Brown/Grey
sex	The sex of the horse	Categorical	Gelding, Mare, Colt
importtype	The import type of the horse	Categorical	PP, PPG, SG
owner	The owner of the horse	Categorical	Zippalanda Syndicate
sire	The sire of the horse	Categorical	Savabeel
dam	The dam of the horse	Categorical	Grand Princess
damsire	The dam's sire of the horse	Categorical	Last Tycoon
url	A url linked to the HKJC data source	/	/
age	The current age of the horse	Real Value	/

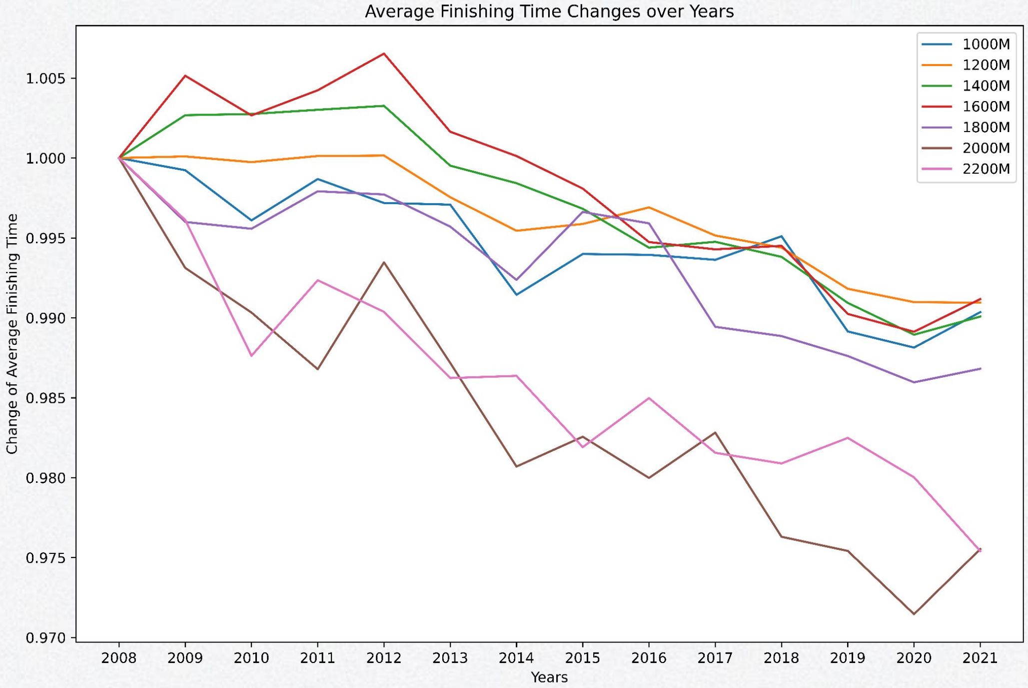
Horse-race Data

Features	Descriptions	Types	Samples
horseid	A unique id of the horse	Categorical	HK_2016_A061
raceid	A unique id of the race	Categorical	1979-09-22-001-1235-Grass
place	The final place of the horse	Categorical	1 – 14, DISQ, DNF
draw	The draw where the horse starts racing	Categorical	Integer with range of 1 - 14
rating	The rating of the horse	Real Value	Decimal with range of 1 – 144
trainer	The trainer of the horse	Categorical	R Gibson
jockey	The jockey riding the horse	Categorical	K H Chan
lengthbehindwinner	The distance between the first placed horse and the horse in the race (in horse length)	Real Value	9-½, 6-¼, 5
winodds	The final WIN odds of the horse	Real Value	/
actualweight	The actual weight of the horse	Real Value	Integer with range of 113 – 133
position{1 – 6}	The place of the horse at different distance intervals	Categorical	1 – 14, DISQ, DNF
finishtime	The finishing time of the horse	Real Value	/
declaredweight	The declared weight of the horse	Real Value	/
gear	The gear equipped by the horse	Categorical	B, TT, B/TT

Betting Odds Data

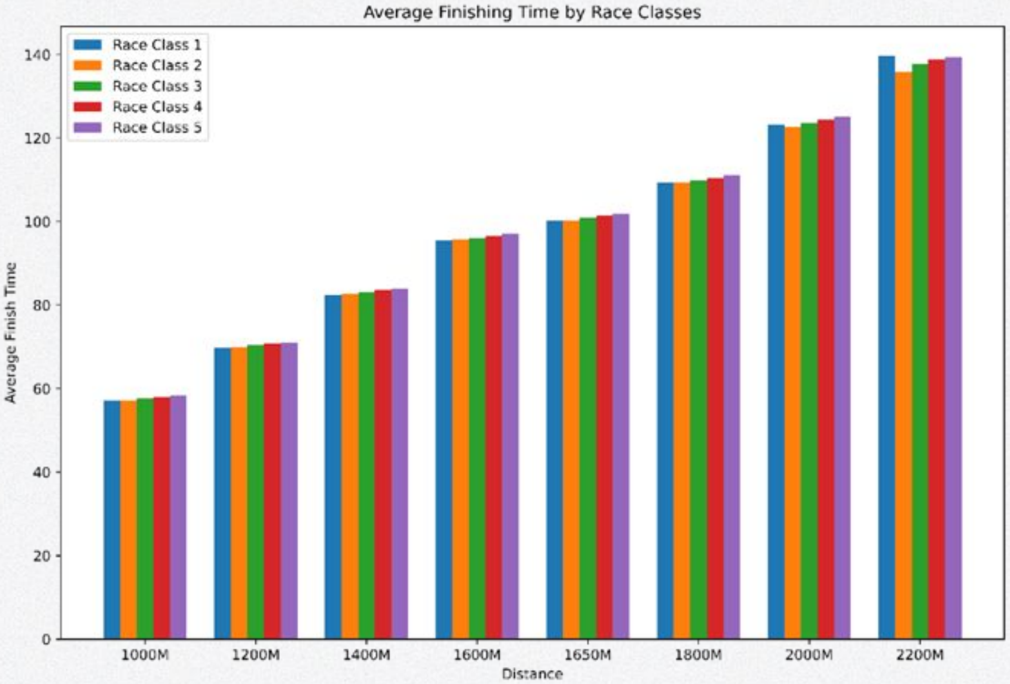
Features	Descriptions	Types	Samples
horseid	A unique id of the horse	Categorical	HK_2016_A061
raceid	A unique id of the race	Categorical	1979-09-22-001-1235-Grass
oddtype	The type of betting odds	Categorical	W, P
odd_1	The betting odds released by the Hong Kong Jockey Club	Real Value	/
odd_{2 – 12}	The betting odds after the release of betting odds (per hour)	Real Value	/
odd_{13 – 28}	The betting odds between 12 hours after release of betting odds and 2.5 hours before the race starts	Real Value	/
odd_{29 – 55}	The betting odds from 2.5 hours to 30 minutes before the race starts (per 5 minutes)	Real Value	/
odd_{56 – 84}	The betting odds in the last 29 minutes (per minute)	Real Value	/
odd_85	The final betting odd	Real Value	/

Data Analysis



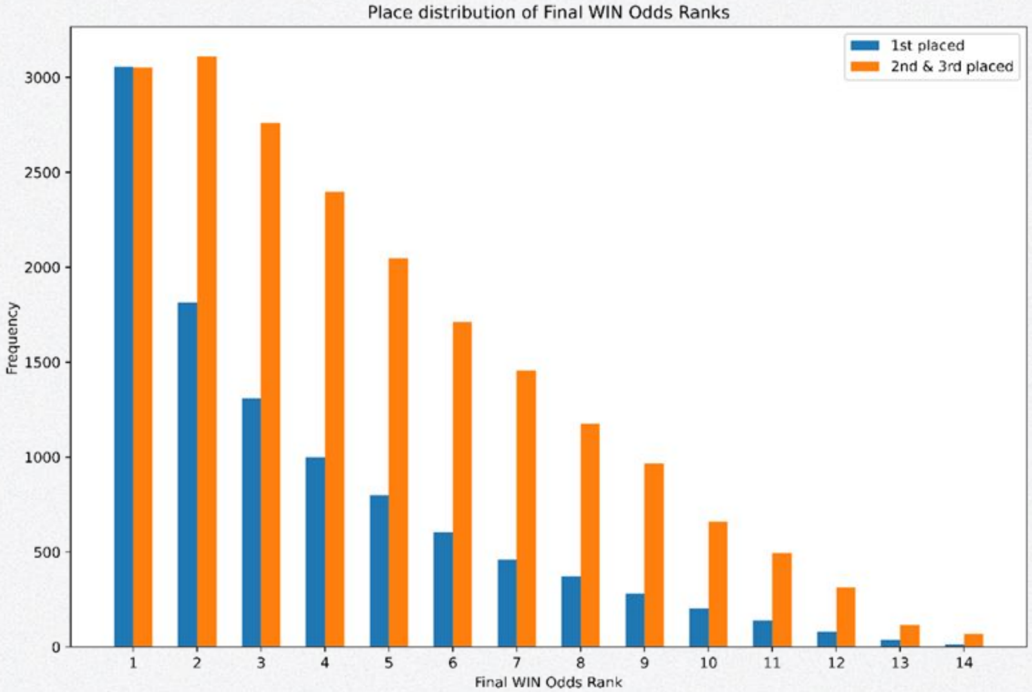
- Decreasing trend in finishing time
- Increasing of performance

Data Analysis



- The performance of different race classes has no significant difference

Data Analysis



- The horses with lower odd have higher chances to win
- The peak of 2nd & 3rd place is rank 2

Data Encoding

- **One-hot encoding**
  - For data where values have no relation
- **Ordinal encoding**
  - For data where values has ordinal relation
- **Customized encoding for gears equipped by horses**
  - Indicate the **experience** of equipping this gear

Encoded variables	Description
0	Not equipped
1	Equipped for one or more consecutive races
2	Equipped for the first time since this race
3	Equipped the gear again since last unequipping the gear
4	Unequipped since this race

Data Imputing

- **Distance interval data (Position data)**
  - Missing value due to different race distance
- **Imputing approach:**
  - Constant value
  - Attribute mean
  - Speed calculated by horse

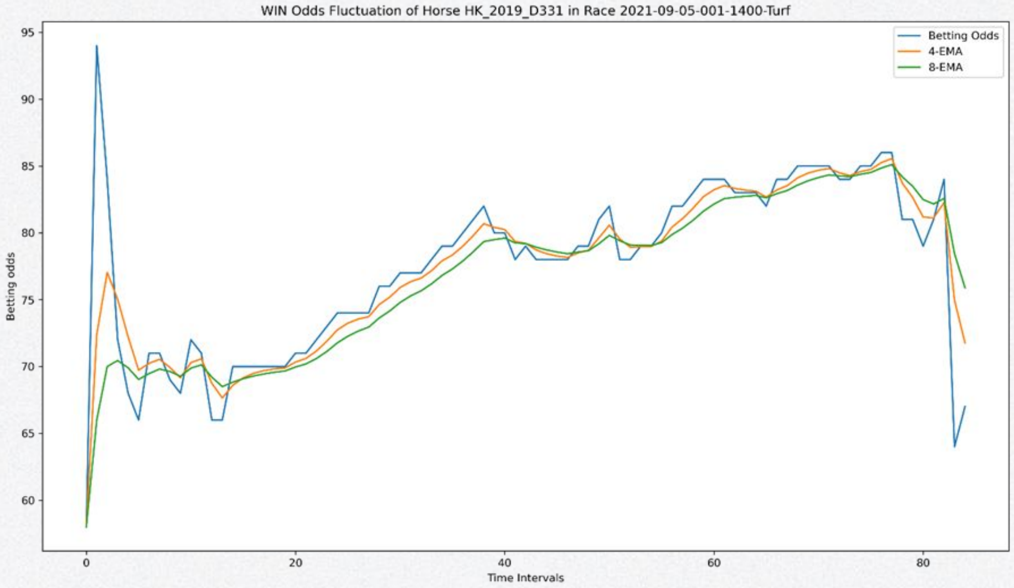
	2400 – 2000M	2000 – 1600M	1600 – 1200M	1200 – 800M	800 –400M	Last 400M
1000M	/	/	/	(1000 – 800M)	✓	✓
1200M	/	/	/	✓	✓	✓
1400M	/	/	(1400 – 1200M)	✓		✓
1600M	/	/		✓	✓	✓
1650M	/	/	(1650 – 1200M)	✓	✓	✓
1800M	/	(1800 – 1600M)	✓	✓	✓	✓
2000M	/	✓	✓	✓	✓	✓
2200M	(2200 – 2000M)	✓	✓	✓	✓	✓
2400M	✓	✓	✓	✓	✓	✓

Additional Features

Features	Descriptions	Types	Sample
weight_diff	The difference of declared weight of the horse between last race and current race	Real Value	Positive or Negative Value
last_weight	The declared weight of the horse in last race	Real Value	Positive Value
last_rating	The rating of the horse in the last race	Real Value	Positive Value
last_place	The final place of the horse in last race	Categorical	1 – 14
count_{1-14}	The count on final place the horse got in the past	Real Value	Positive Value
last_pos_time_{1-6}	The time of the horse at different distance interval in last race	Real Value	Positive Value
last_pos_place_{1-6}	The place of the horse at different distance interval in last race	Categorical	1 – 14
last_speed	The average speed of the horse in the last race	Real Value	Positive Value
avg_rating	The average rating of the horse in all previous races	Real Value	Positive Value
avg_pos_time_{1-6}	The average time of the horse at different distance interval in all previous races	Real Value	Positive Value
avg_pos_place_{1-6}	The average place of the horse at different distance interval in all previous races	Real Value	Positive Value less than 14
avg_speed	The average speed of the horse in all previous races	Real Value	Positive Value
avg_finishtime	The average finishing time of the horse in all previous races	Real Value	Positive Value
win_odds_rank	The ranks of final win odds in the races	Categorical	1 – 14

- Previous performance
  - From last race
  - Average in career
- Win odds ranking
  - Relation of horses in race

Additional Features



Betting Odds Data

- Exponential Moving Average (EMA)
    - Display underlying trend
    - Indicate the significant changes / trend breaking
    - Only for last 10 minutes

3

Horse Racing Prediction

Procedure, Evaluation & Performance

Input Data for Training

Features	Types	Encoding Methods
raceclass	Categorical	Ordinal
tracktype	Categorical	One-hot
racktrack	Categorical	One-hot
course	Categorical	One-hot
country	Categorical	One-hot
importtype	Categorical	One-hot
sex	Categorical	One-hot
colour	Categorical	One-hot
going	Categorical	One-hot
jockey	Categorical	Ordinal
trainer	Categorical	Ordinal
horseid	Categorical	Ordinal
dam	Categorical	Ordinal
sire	Categorical	Ordinal
damsire	Categorical	Ordinal
distance	Categorical	Ordinal
draw	Categorical	Ordinal
rating	Real Value	/
rating_rank	Real Value	/
last_rating	Real Value	/
avg_rating	Real Value	/
last_place	Real Value	/
winodds	Real Value	/
win_odds_rank	Real Value	/
actualweight	Real Value	/
declaredweight	Real Value	/
gear	Categorical	Customized Encoding
raceidseason	Real Value	/
count_{1 – 3}	Real Value	/
weight_diff	Real Value	/
avg_finishtime	Real Value	/
avg_pos{1 – 6}_pos	Real Value	/
avg_pos{1 – 6}_time	Real Value	/
last_pos{1 – 6}_pos	Real Value	/
last_pos{1 – 6}_time	Real Value	/

- Features included
  - Races data
  - Horses data
  - Horse-race data
  - Additional features
- Drop unnecessary , irrelevant features
- Split train and test data according to race season
  - **Training** data: **2008 - 2019**
  - **Testing** data: **2019 - 2021**

Model Configurations

- **Optimized by cross-validated grid-search**
    - No. of decision trees: **256**
    - Max. depth of decision tree: **13**
    - Min. no. of samples required to split an internal node: **2**
    - Metric for comparing the quality of each node split: **mean squared error**

Results and Analysis

Results of Random Forest

	horseid	raceid	distance	winodds	place	pred
17995	5,265.00	2019-09-01-001-1600-Turf	3.00	2.20	1	95.84
17993	4,296.00	2019-09-01-001-1600-Turf	3.00	7.00	5	95.85
17994	4,268.00	2019-09-01-001-1600-Turf	3.00	5.70	4	95.86
17996	5,186.00	2019-09-01-001-1600-Turf	3.00	4.90	2	95.89
17999	4,302.00	2019-09-01-001-1600-Turf	3.00	18.00	3	96.16
18000	4,982.00	2019-09-01-001-1600-Turf	3.00	21.00	9	96.17
18001	4,809.00	2019-09-01-001-1600-Turf	3.00	19.00	6	96.21
17998	4,845.00	2019-09-01-001-1600-Turf	3.00	14.00	8	96.33
17997	5,103.00	2019-09-01-001-1600-Turf	3.00	50.00	7	96.35

- Sample output of a race
- Small difference between predicted time

Decision Path

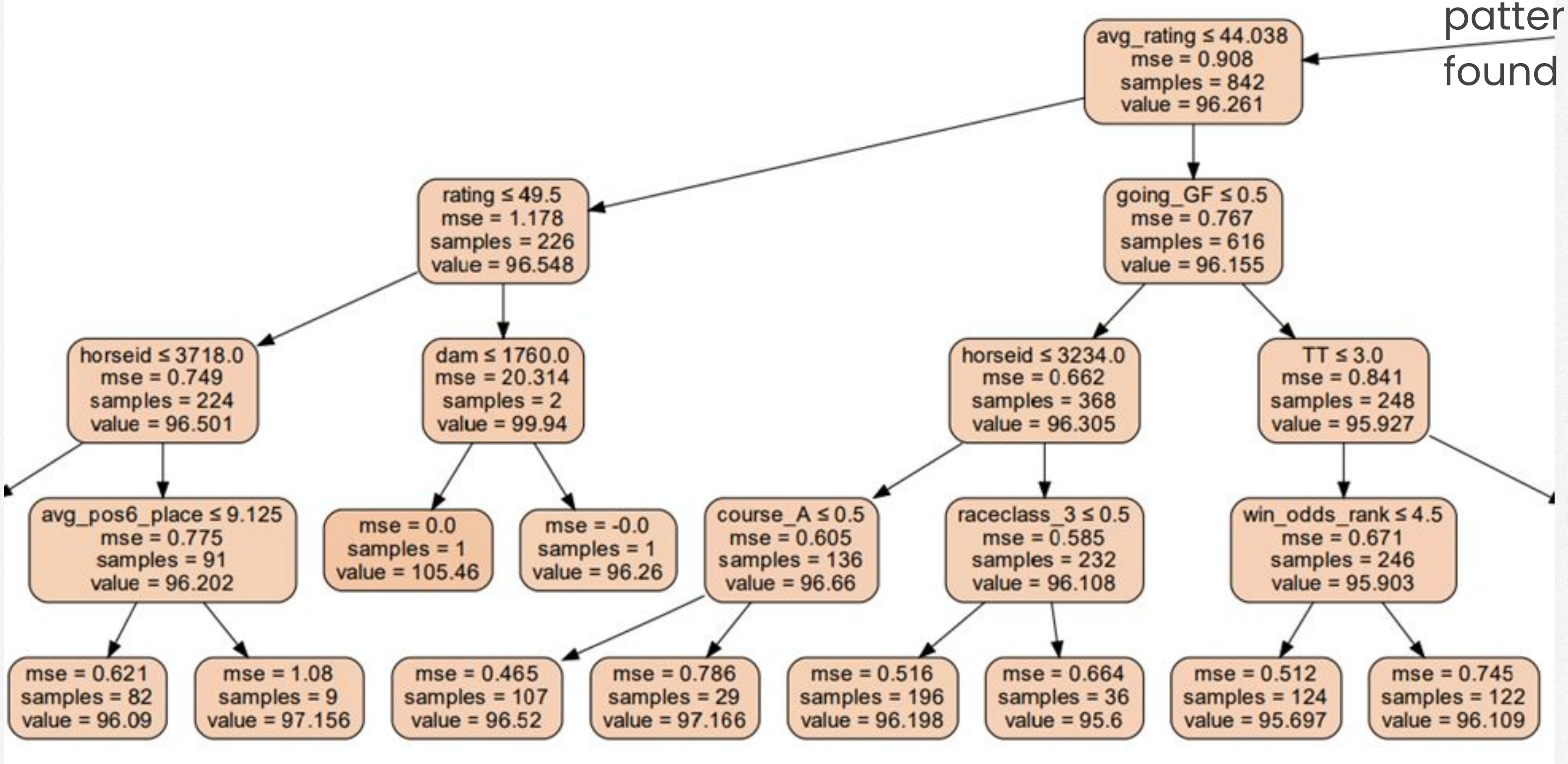
	avg_finishtime	avg_pos6_place	avg_pos6_time	avg_rating	declaredweight	distance	draw	going_GY	horseid	raceidseason	rating	win_odds_rank
17994	96	4.2	23	36	1.1e+03	3	3	0	4.3e+03	1	37	3

```
decision id node 0 : (X_test[1, 'distance'] (= 3.0) > 2.5)
decision id node 3982 : (X_test[1, 'distance'] (= 3.0) <= 5.5)
decision id node 3983 : (X_test[1, 'distance'] (= 3.0) <= 4.5)
decision id node 3984 : (X_test[1, 'distance'] (= 3.0) <= 3.5)
decision id node 3985 : (X_test[1, 'rating'] (= 37.0) <= 63.5)
decision id node 3986 : (X_test[1, 'win_odds_rank'] (= 3) <= 9.5)
decision id node 3987 : (X_test[1, 'horseid'] (= 4268.0) > 2408.0)
decision id node 4075 : (X_test[1, 'going_GY'] (= 0.0) <= 0.5)
decision id node 4076 : (X_test[1, 'raceidseason'] (= 1) <= 688.5)
decision id node 4077 : (X_test[1, 'avg_rating'] (= 36.0) <= 44.038461685180664)
decision id node 4078 : (X_test[1, 'rating'] (= 37.0) <= 49.5)
decision id node 4079 : (X_test[1, 'horseid'] (= 4268.0) > 3718.0)
decision id node 4083 : (X_test[1, 'avg_pos6_place'] (= 4.166666666666667) <= 9.125)
```

- **Actual** finishing time: **95.68**
- **Predicted** finishing Time: **96.09**
- **Distance** are the first features for node splitting
- **odds**, **rating**, **going** are used as node splitting features

Structure of Decision Tree

- Small size leaves
  - Not overfitting
  - Underlying pattern is found



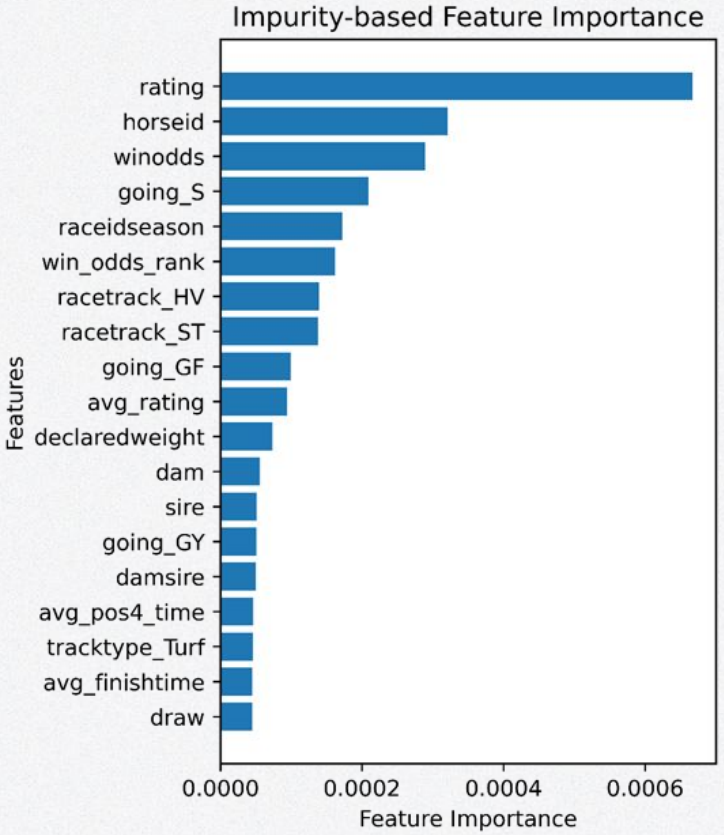
Evaluation Metrics

- **Mean Squared Error (MSE)**
    - Accuracy of the prediction
    - Closer to 0, the better performance
    - **MSE of model: 2.2649 seconds**
- **Explained Variance Score**
    - Discrepancy between the model and data
    - The closer to 1, the stronger association
    - **Explained Variance Score of model: 0.99388**

Features Importance

- **Impurity-based Feature Importance**
  - Measure the significance of affecting decision trees
  - Related to node splitting
  - Computed by
    - Mean and standard deviation of **accumulation of the impurity reduced**
    - Take average among all decision trees
  - **Limitation:**
    - Misleading when features have high cardinality

Features Importance

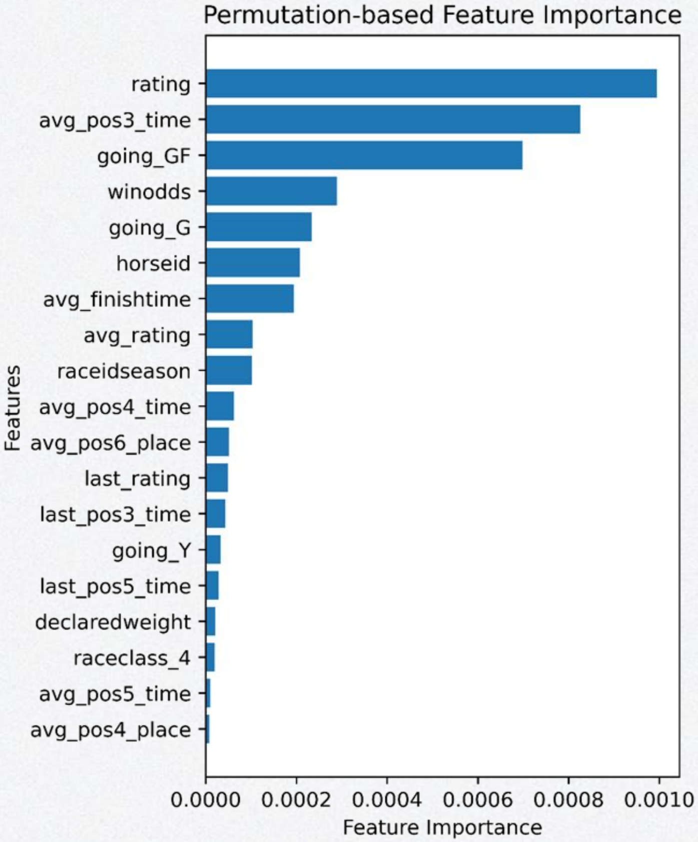


- Top 20 features
- **Distance** is hidden
  - Significantly greater than other features
  - **Importance: 0.99631**
- **Rating** and **odds** occupied **4/20**
  - Performance & public intelligence
- **Environmental** features occupied **6/20**
  - Racetrack
  - Going
  - Tracktype
  - Raceidseason
- High cardinality features

Features Importance

- **Permutation-based Feature Importance**
  - Measure the significance of affecting the prediction result
  - Computed by
    - Define a **baseline metric** (R^2 score)
    - Evaluate the model on given dataset
    - **Permute** each feature from the dataset and evaluate the model with the same metric
    - Figure out the **difference** between the baseline metric and newly computed metric
  - **Limitation:**
    - Misleading when there are highly correlated features

Features Importance



- Top 20 features
- **Distance** is hidden
  - Significantly greater than other features
  - **Importance: 2.0142**
- **Rating** and **going** have high importance
- Id of horse has been **verified**
- More distance interval data
  - Previous performance affect the result of prediction

Betting Simulation

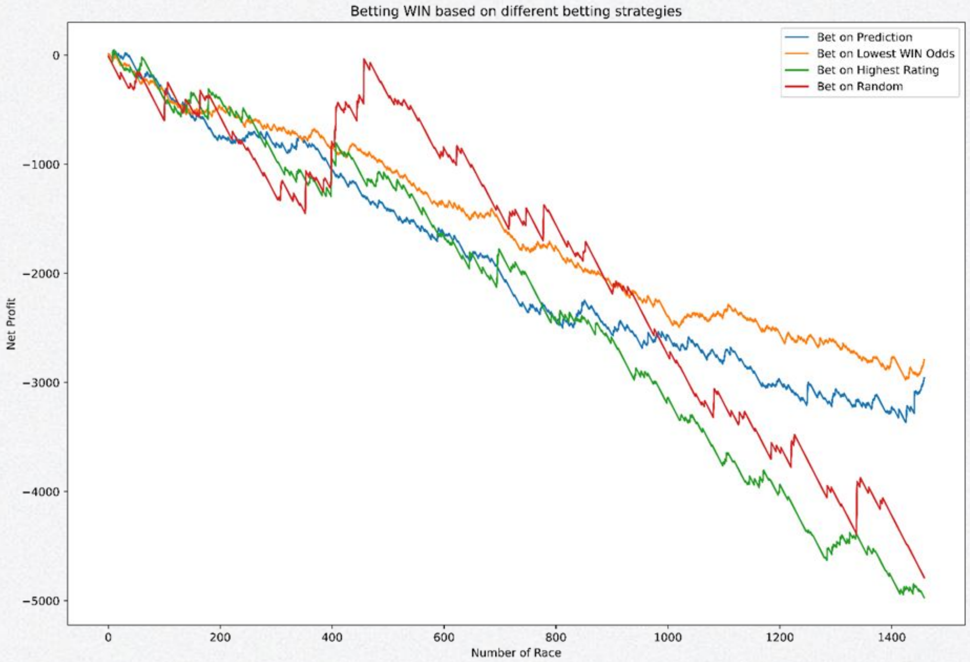
1. Group all the horses by the race
2. Order the horses by the predicted finishing time in ascending order
3. Assign a **predicted place** to each horse according to the ranking
4. Start Betting!

	horseid	raceid	place	winodds	pred	pred_place	place_difference
17995	5265.0	2019-09-01-001-1600-Turf	1	2.2	95.84	1	0
17996	5186.0	2019-09-01-001-1600-Turf	2	4.9	95.89	4	2
17999	4302.0	2019-09-01-001-1600-Turf	3	18.0	96.16	5	2
17994	4268.0	2019-09-01-001-1600-Turf	4	5.7	95.86	3	-1
17993	4296.0	2019-09-01-001-1600-Turf	5	7.0	95.85	2	-3
18001	4809.0	2019-09-01-001-1600-Turf	6	19.0	96.21	7	1
17997	5103.0	2019-09-01-001-1600-Turf	7	50.0	96.35	9	2
17998	4845.0	2019-09-01-001-1600-Turf	8	14.0	96.33	8	0
18000	4982.0	2019-09-01-001-1600-Turf	9	21.0	96.17	6	-3

Betting Simulation

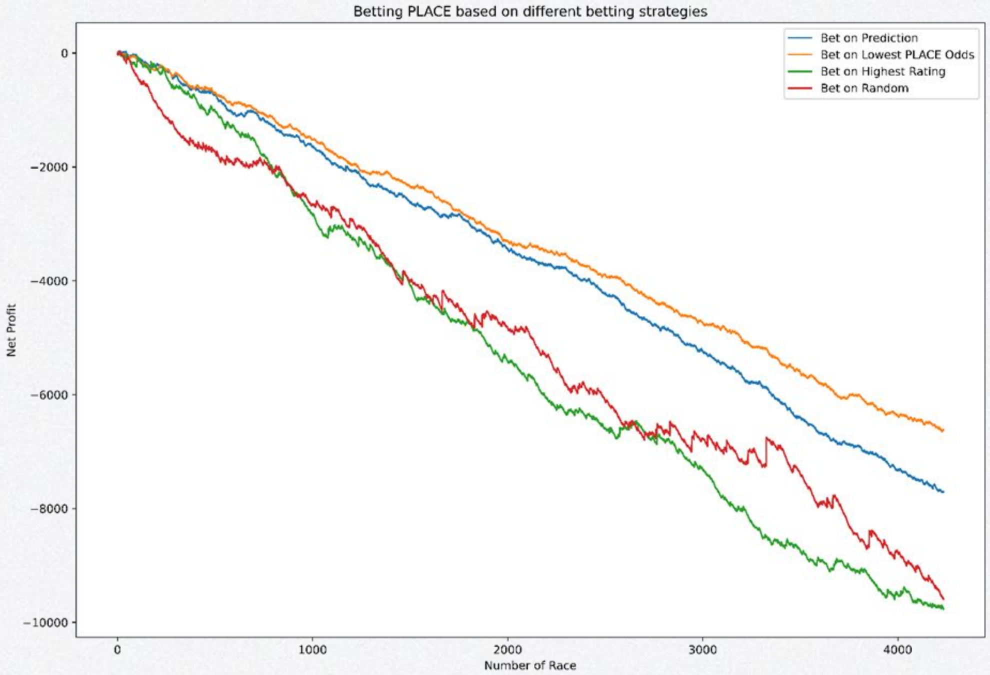
1. Assume $10 would be used for each bet
2. Gain **$10 * odds – 10** if correctly picked the horses
3. Lose $10 otherwise
4. **PLACE** betting would be simulated
5. Compare with different strategies
   - Based on lowest odds
   - Based on highest rating
   - Random

Betting Simulation



- Betting **WIN**
- All strategies are **losing** money
- Random & rating have the **worst** performance
- Comparable result with betting on lowest odds
  - **Lowest odds has 30% accuracy**
  - Our prediction has **24.537%**

Betting Simulation



- Betting **PLACE**
- All strategies are **losing** money
- Random & rating have the worst performance
- Comparable result with betting on lowest odds
  - **Lowest odds has 51.349% accuracy**
  - Our prediction has **47.153%**

4

Horse Betting Using MAB

MAB Formulation of Horse Betting Problem

- **Contextual MAB problem**
  Each action (horse) is with context (such as closing odds), which is correlated to the reward (outcome of the bet)
- **Combinatorial MAB problem**
  Instead only one action is played at a time, multiple actions are played (Placing multiple bets) at once.

Contextual MAB

Possible features (context) for horse betting:

- First/closing odds, intermediate odds
- Rankings
- etc…

How does these features relate to the outcome (how much/likely will we earn)?

- Linear relation can be a reasonable guess

Examples for algorithms with linear model: **LinUCB, LinTS**

Combinatorial MAB

In many cases, we would want to play **multiple actions** instead of one only

- Article/Ad recommendation
- Allocate jobs to multiple workers in crowdsourcing

True also in our case, it would be **more flexible** if we allow **betting on multiple horses**

Bet type we play: **Place bet**
Max horses to bet: **2**

Algorithms

Some of algorithms that we use: **LinUCB**[1], **LinTS**[2]
-> For one-armed bandit problem (one action each round)
To extend from that, we just pick multiple actions with highest scores.

Combinatorial MAB algorithms we tried:
CC-MAB[3], **C²UCB**[4] (not included as not working well)

Environment for Horse Betting

We will let the agent play race by race

Action set:

14 horses (at most) ordered by predicted finishing time + not to bet

Features (for each horse):

- Last moment place odds
- Last 10 minutes EMA of odds
- Rankings (odds, predicted finishing time)
- Ratio of finishing time between each horse with the horse ranked 1 place ahead (finishing time)

Environment for Horse Betting(cont.)

Reward functions

1. Bet right or wrong (Bet any of top 3 horses correctly) = 1
   - R(Bet wrong) = 0
   - R(Not bet) = 0.46,   i.e. average place accuracy

->Expected to pick actions that win most

2. Actual cash change
   - R(Bet any of top 3 horses correctly) = 10 * place odd - 10 (cost)
   - R(Bet wrong) = -10
   - R(Not bet) = -3, i.e. average return of random guess

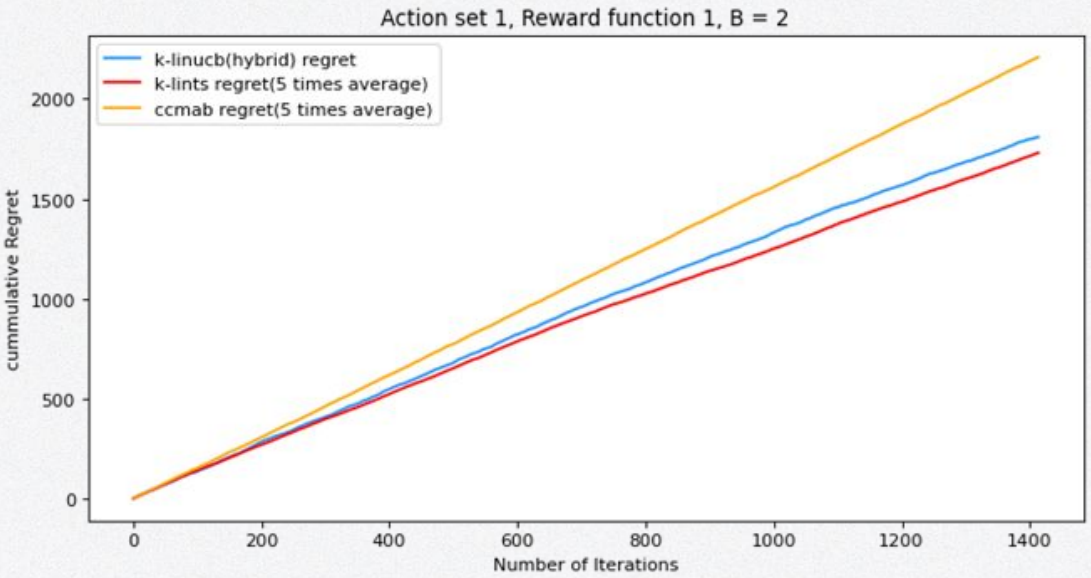
->Expected to pick actions that earns the most

Procedure

- **Dataset**
  - 1414 races in total
    - No train-test split (bandit algo are online)
    - Run according to chronological order
    - 6-14 horses in each

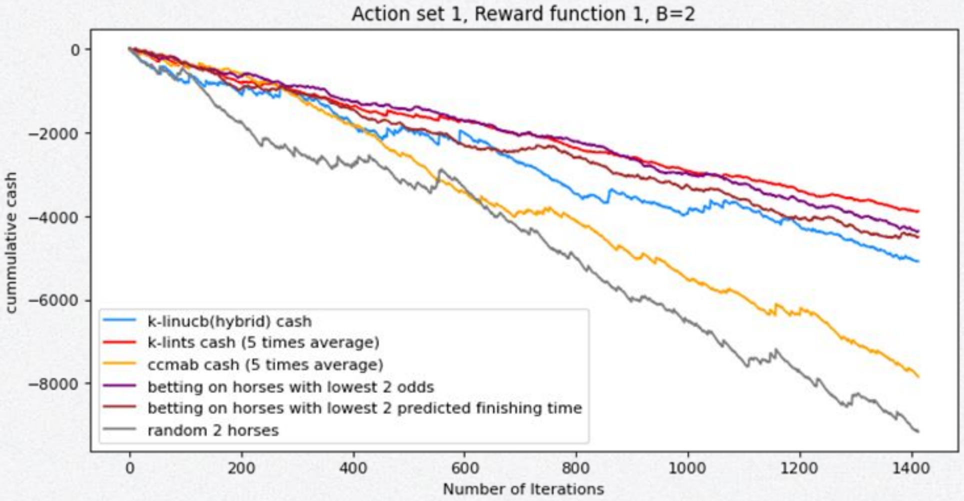
Results and Analysis

Regret (Reward Function 1)



Regret slightly improves Overtime
->Agent does learn to take better actions as time goes on

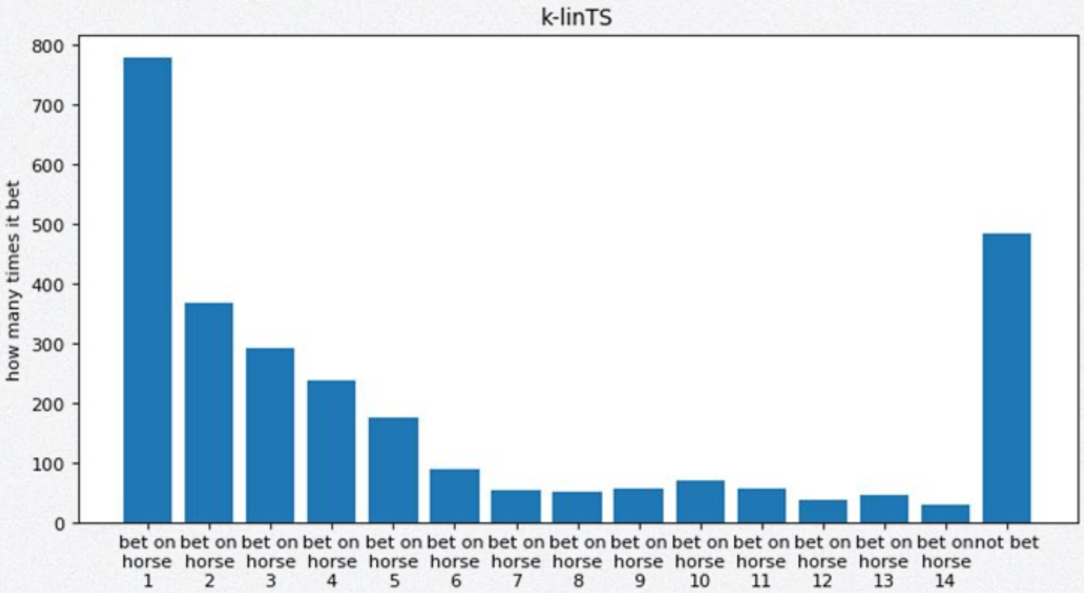
Cash Balance (Reward Function 1)



Even we just let the agent learns on the go (no training), the result is still comparable to public intelligence/our prediction in terms of

- Descent rate of cash
- Final balance

Actions Taken (Reward Function 1)



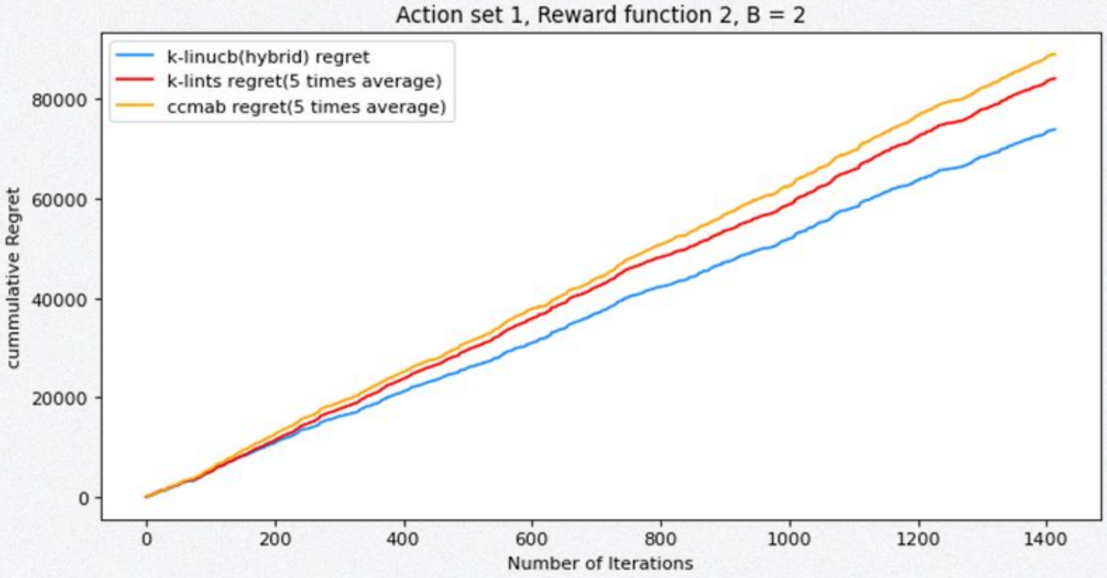
The agent is able learn to play Horses with highest win rate (horse 1)

Result (Reward Function 1)

The agent **can** learn in such simple scenario of guessing which actions win the most (as reward function 1 favors horses with high win rate).

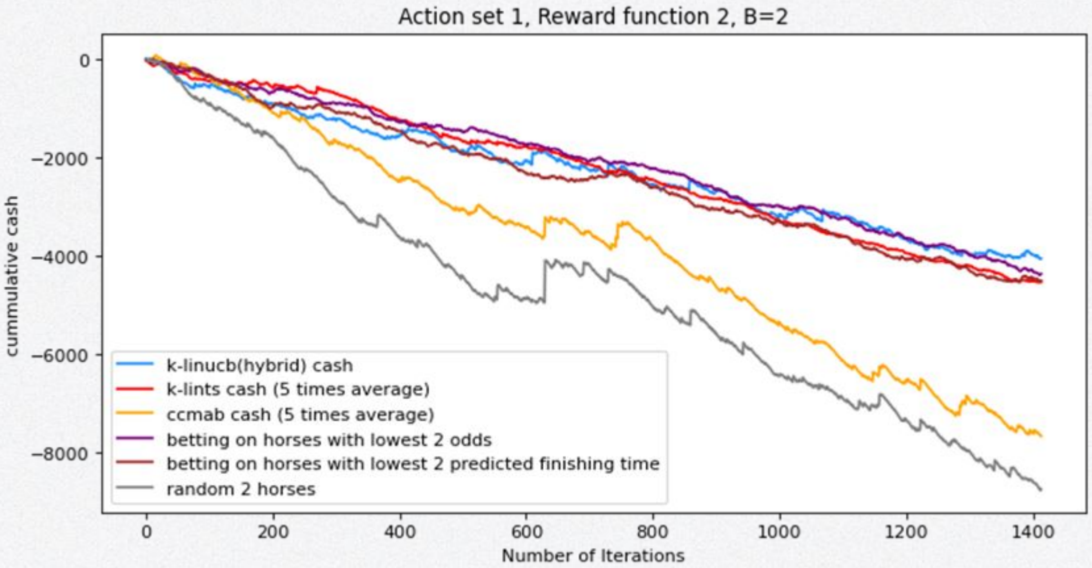
But how about more sophisticated scenario where we really use actual money gain/lose as reward (reward function 2)?

Regret (Reward Function 2)



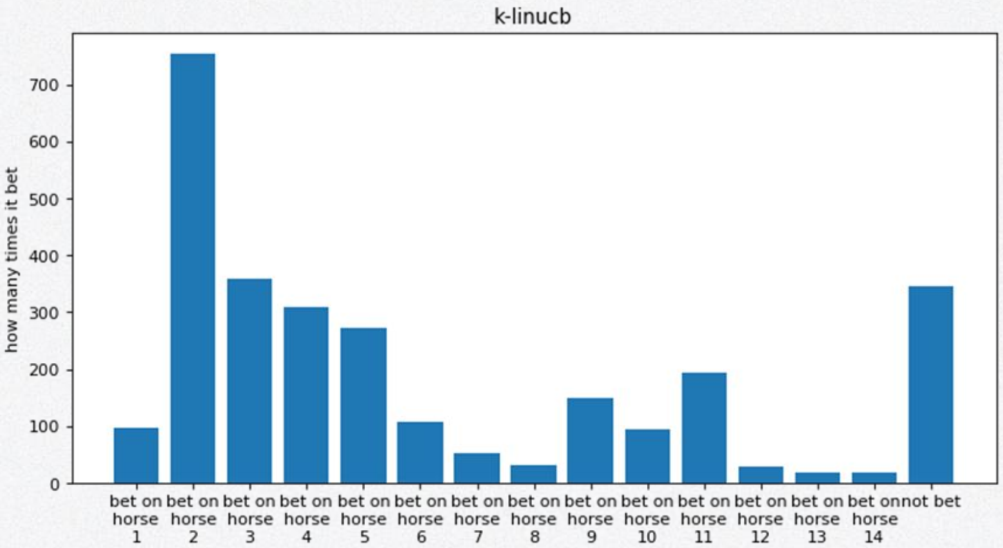
Compared to last time, the regret can hardly reduce

Cash Balance (Reward Function 2)



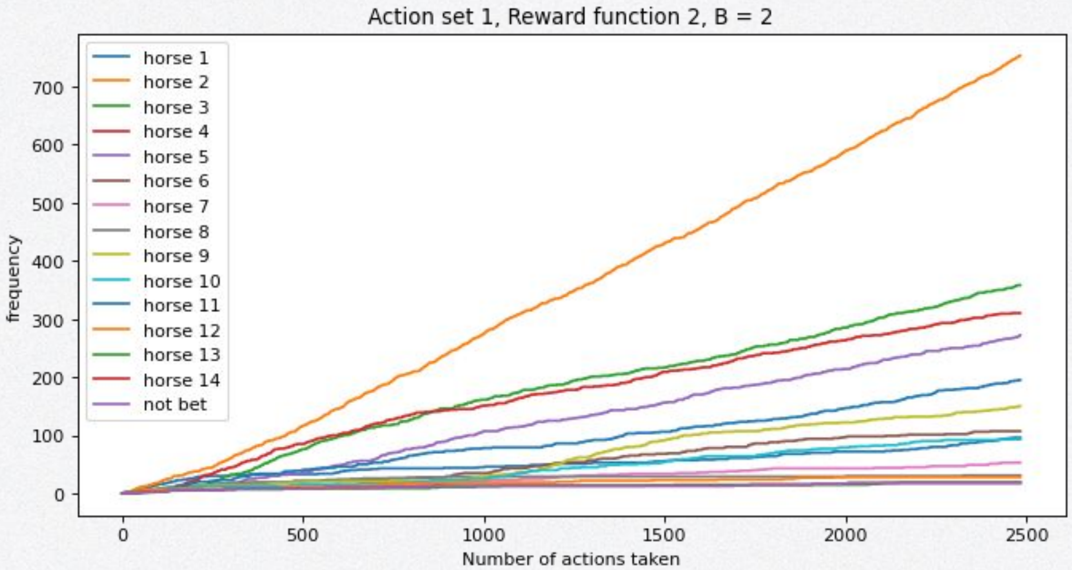
- earns a bit more than public intelligence/our prediction
- Still cannot revert the trend of losing money.

Actions Taken (Reward Function 2)



- **Most frequent actions**
  - Horse 2-5
  - Higher odds but still likely to be in top 3
- But still not sufficient to generate profit

Actions Taken (Reward Function 2)



As time goes on, horse 2-5 are played significantly more than other horses.

Result (Reward Function 2)

- Able to choose actions with higher profit (not just simply choose top 3)
- Perform a bit better than public intelligence/our prediction
- Still insufficient for generating profit
  - All options have negative average return (low odds & accuracy not high enough)
  - Fixed bet each time

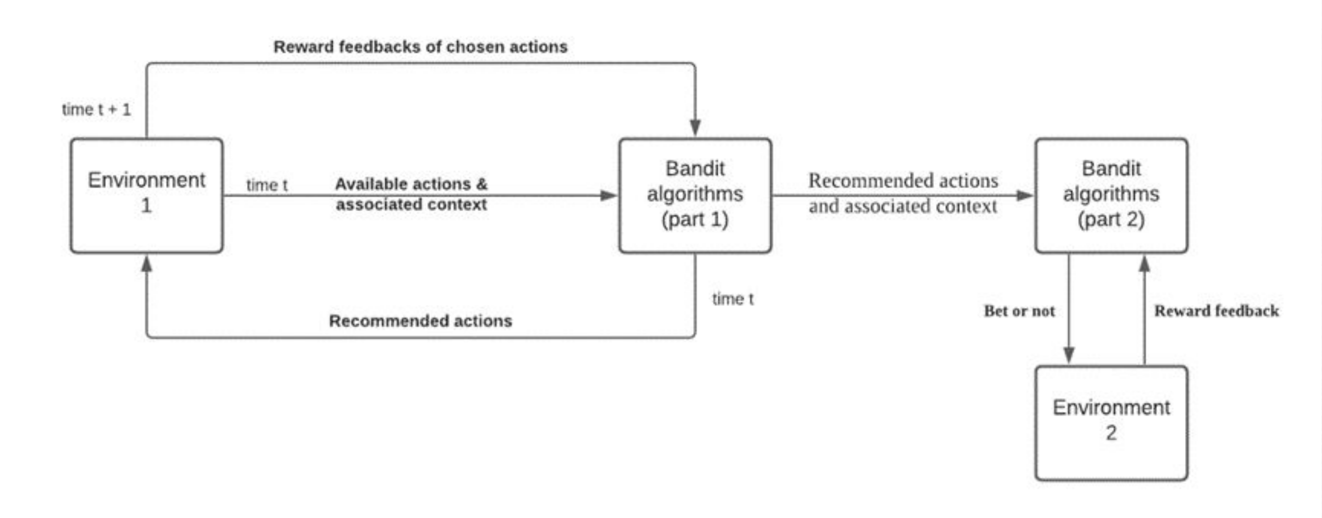
Possible Improvements

- Previously we bet fixed amount of money,
  Can we bet more especially when it could possibly earn decent amount (e.g. >= 20)?

- And if the money returned can be low (e.g. < $15), we don't earn much anyway. Do we still continue to bet?

Our attempt: use 2 extra bandit algorithms to make decision for each situation

Procedure



- Feed actions from original bandit to 2 extra bandits
    - One for each situation in last slide
    - Output ‘bet’ or ‘not bet’ actions

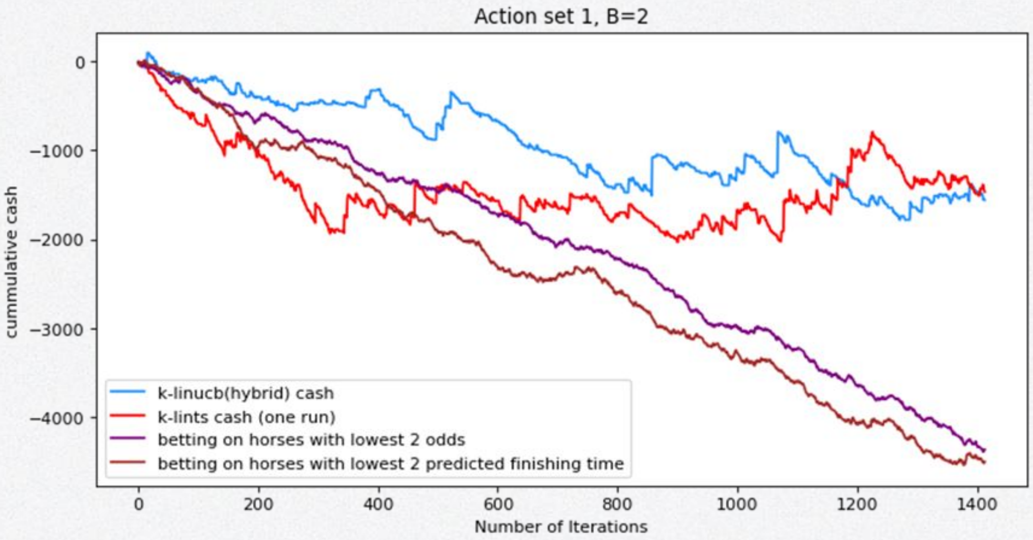
Environment (Part 2)

Reward functions

1. For 1st bandit (bet when possible return is >=15)
   - R(Bet and return >= 15) = return
   - **R(Bet and return < 15) = -10**
   - **R(Not bet and return when bet < 15) = 10**
   - R(Not bet and return when bet >= 15) = -net gain if bet

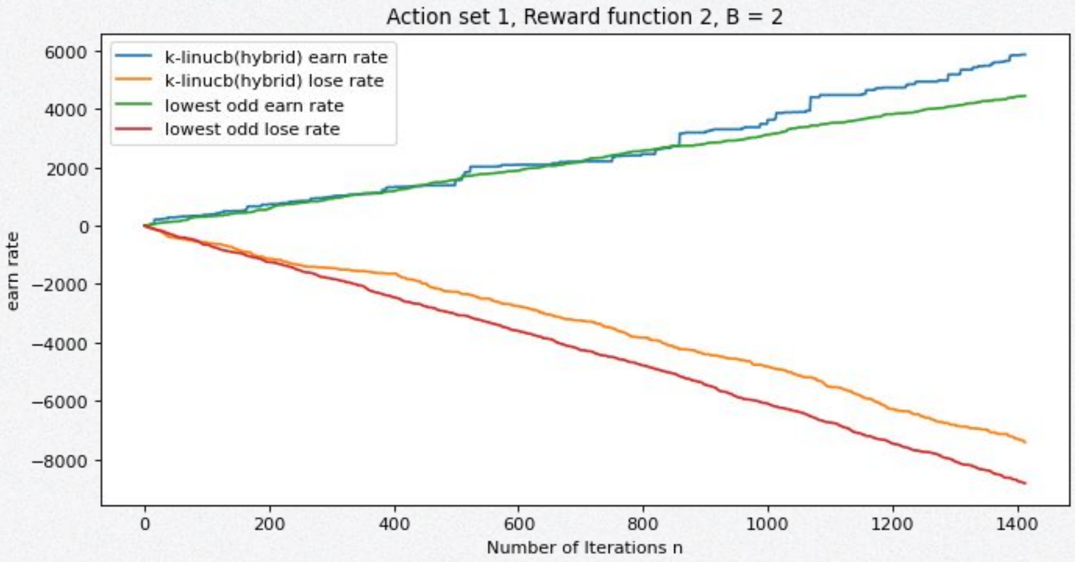
2. For 2nd bandit (bet $20 when possible return is >= 20)
   - R(Bet and return >= 20) = return *2
   - **R(Bet and return < 20) = -20**
   - **R(Not bet and return when bet < 20) = 10**
   - R(Not bet and return when bet >= 20) = -net gain if bet $10 *2

Results



- make large gains oftenly
- **Red** curve:
  - doesn't lose much after certain point
  - large ascent at the end.

Results



The frequency of earning / losing of **k-LinUCB** is **better** than **public intelligence**.

5

Conclusion

Conclusion

- Horse racing prediction model
  - Acceptable accuracy
  - Comparing to the previous projects
  - Predict the result with features in different aspects
    - Odds, rating and environment related
    - Proved to be significant
- Bandit algorithms
  - Capable of finding choices that earn most
  - All choices have negative returns
    - Low odds
    - Hardly have good accuracy
  - Not able generate profit easily
  - Need more advanced way that allows variable amount of money to bet / intelligent enough to decide when to not bet

6

Q&A Section

The End
Thanks!

References

[1]. Li, Lihong, et al. “A Contextual-Bandit Approach to Personalized News Article Recommendation.” Proceedings of the 19th International Conference on World Wide Web - WWW '10, 2010, https://doi.org/10.1145/1772690.1772758

[2]Agrawal, Shipra, and Navin Goyal. "Thompson sampling for contextual bandits with linear payoffs." International Conference on Machine Learning. PMLR, 2013.

[3]Chen, Lixing, Jie Xu, and Zhuo Lu. "Contextual combinatorial multi-armed bandits with volatile arms and submodular reward." Advances in Neural Information Processing Systems 31 (2018): 3247-3256.

[4]Qin, Lijing, Shouyuan Chen, and Xiaoyan Zhu. "Contextual combinatorial bandit and its application on diversified online recommendation." Proceedings of the 2014 SIAM International Conference on Data Mining. Society for Industrial and Applied Mathematics, 2014.