Betting Odds Calculation with Machine Learning

LYU 2102

NAM Man Leung

supervised by Prof. Michael Lyu

Outline

- Introduction
- Background Knowledge
- Data Preparation
- Methodology
- Experiment and Result
- Conclusion

Introduction - Motivation

- Revenue from horse racing is approximately HK$280 billion in 2020-2021 despite the economic downturn caused by the coronavirus pandemic
- Win odds enhance prediction accuracy as shown in previous FYP[1][2]
- Win odds keep changing before the start of the race
- Use machine learning methods to resemble the effect of winning odds in horse racing prediction

Introduction - Objective

- Apply statistical models(rating systems) to evaluate the performance of horses
- Apply techniques in natural language processing for winning horse classification
- Reproduce the effect of variable win odds from the Hong Kong Jockey Club in horse racing prediction by invariable features

Background Knowledge – Rating System

Glicko Rating System[3]

- Rating
  - Performance of a horse
- Rating deviation
  - Reliability of a horse’s rating
  - a low value of rating deviation indicates that the horse joins races frequently and the rating is more reliable
  - uncertainty of a horse’s ability reduces because more information is obtained when the horse joins more races

Background Knowledge – Rating System

TrueSkill Rating System[4]

- Rating
  - Performance of a horse
- Rating deviation
  - Reliability of a horse's rating
- Multiple horse environment
  - Assume outcome of each race is a permutation of multiple horses
  - Allow horses to have the same rank

Background Knowledge – Rating System

Elo-MMR Rating System[5]

- Rating
  - Performance of a horse
- Rating deviation
  - Reliability of a horse's rating
- Multiple horse environment
  - Assume outcome of each race is a permutation of multiple horses
  - Assume horses have distinct ranks
- incentive compatible
  - horses' ratings should not have opposite changes to their performance

Background Knowledge – Transformer

Self Attention mechanism[6]

- Features in the sequence interact with each other
- Assign weights to features according to the relative importance
- Decide dependency relationships between features of the sequence

Data Preparation - Collection

- Write web crawlers by using BeautifulSoup library in python
- Collect data on the HKJC official websites
- Obtain 9191 race records in our dataset dated from June 2008 to October 2021
- Obtain horse records of 6642 horses which participated in those 9191 race records

Data Preparation – Feature Analysis

- By Bayes' formula,

  Pr(Y = win | X = x1, x2, …)

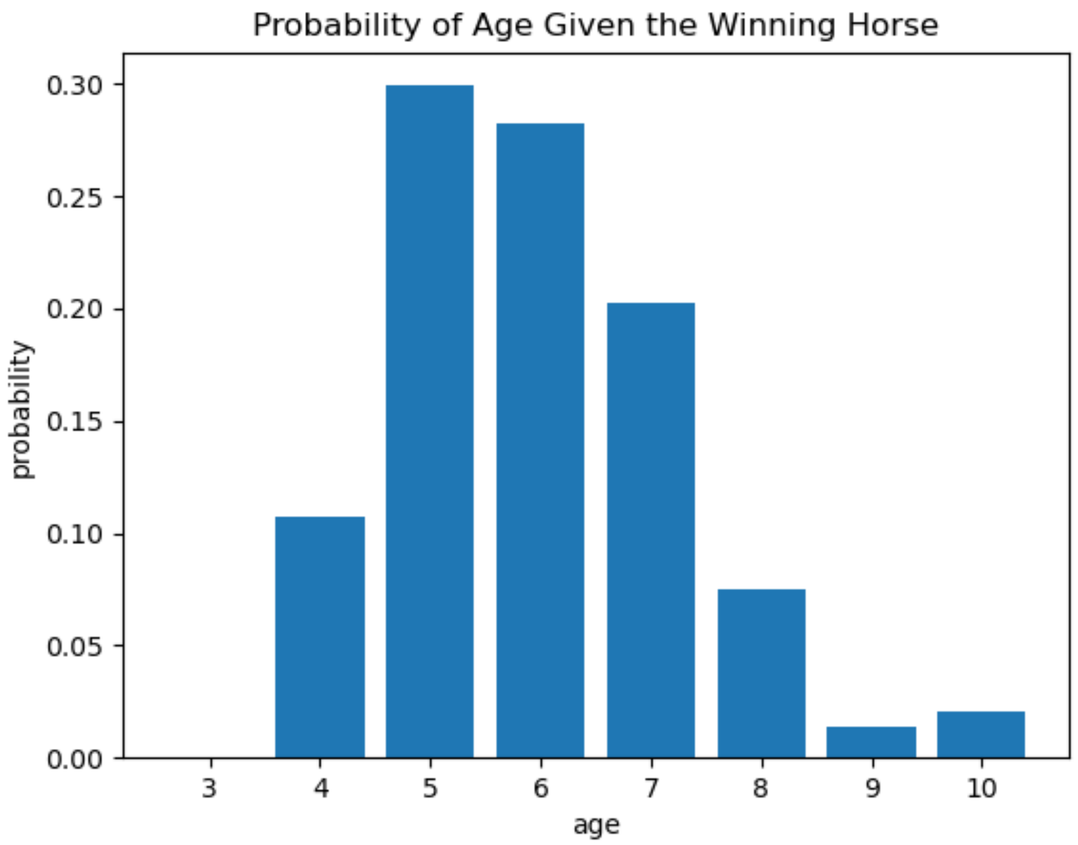
  = Pr(X = x1, x2, … | Y = win) Pr (Y = win) / Pr (X = x1, x2, …)

- Likelihood estimation in machine learning is simplified by assuming that features are conditional independent.

- We observe the likelihood Pr(X = x | Y = win) one by one.

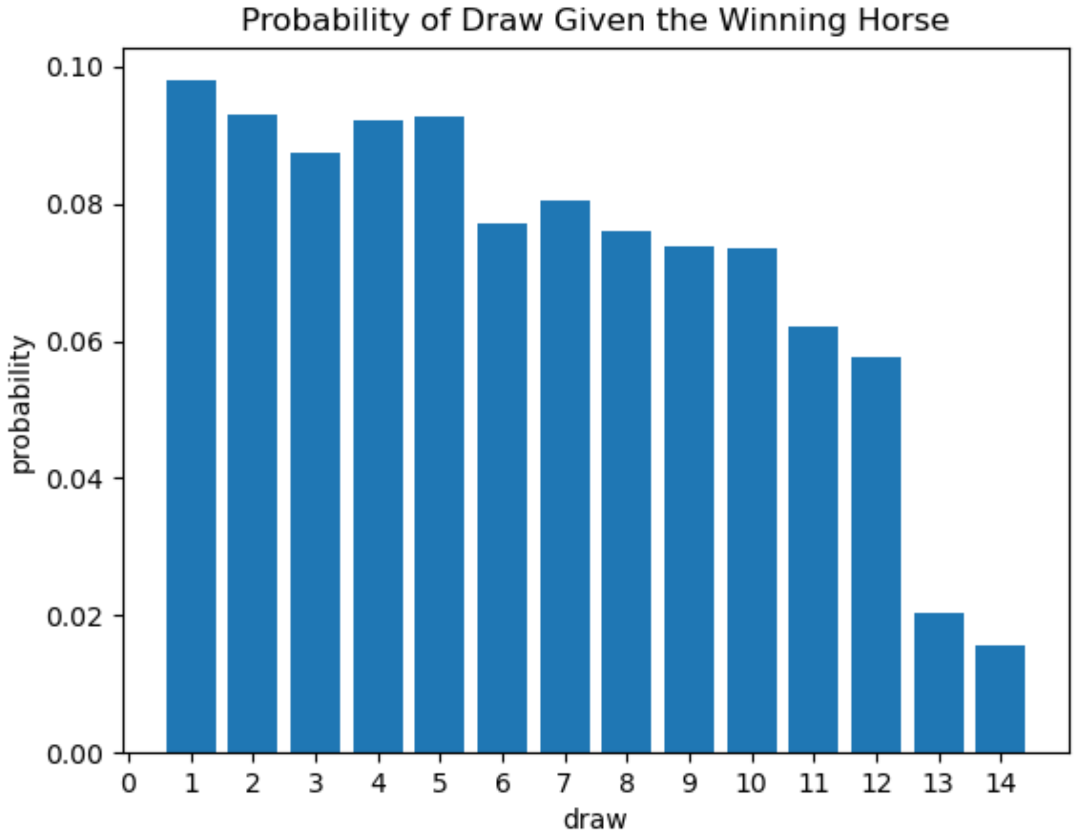
Data Preparation – Feature Analysis (Age)

- Most winning horses are aged between 5 and 6



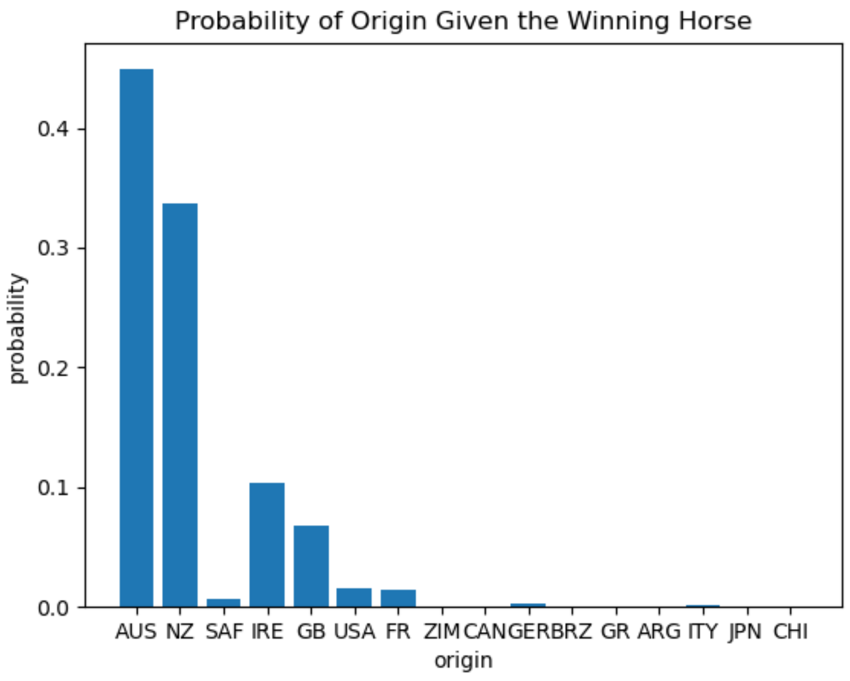
Data Preparation – Feature Analysis (Draw)

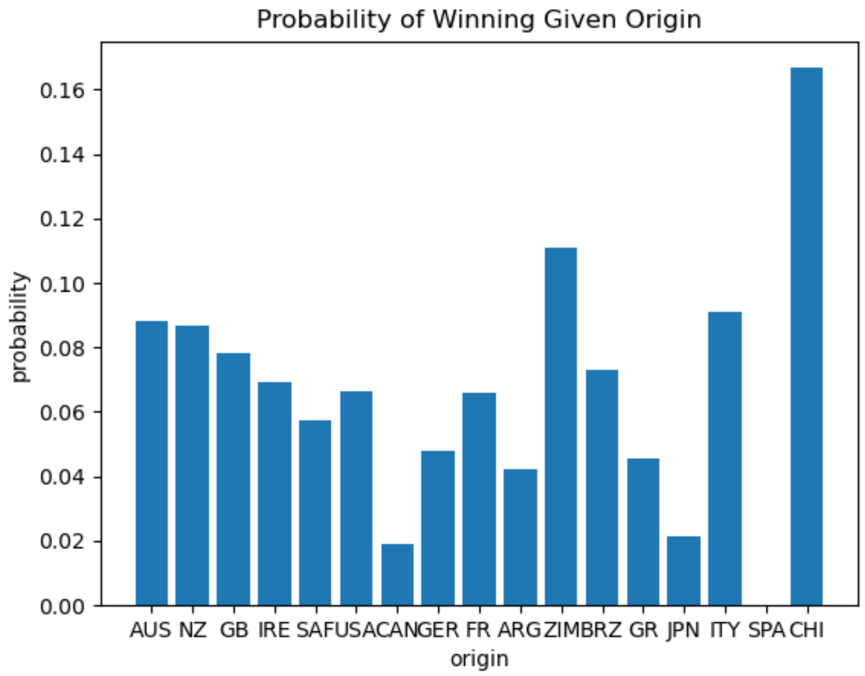
- Most winning horses have smaller draw number.



Data Preparation – Feature Analysis (Origin)

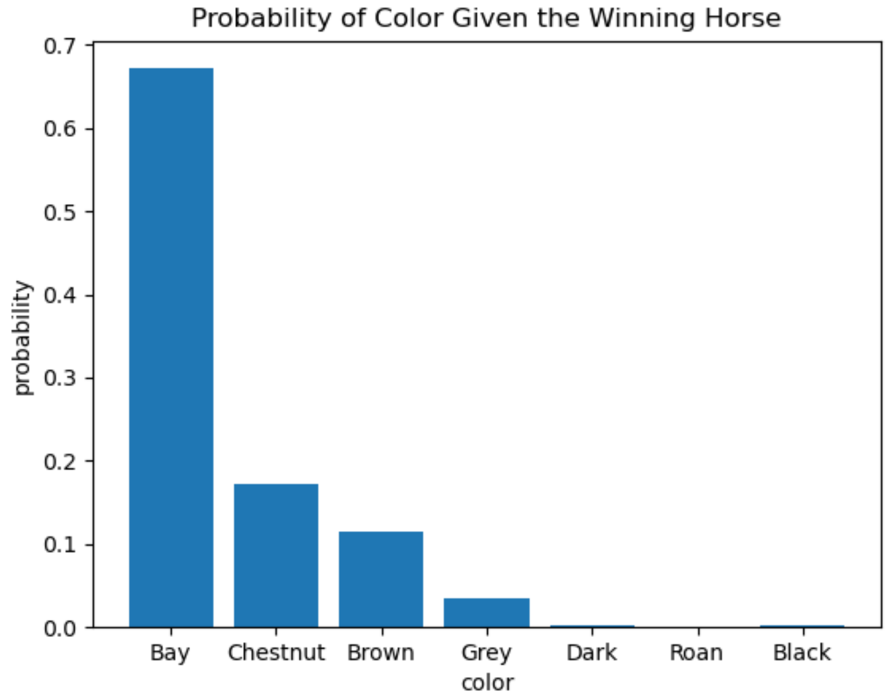
- Most winning horses come from Australia or New Zealand

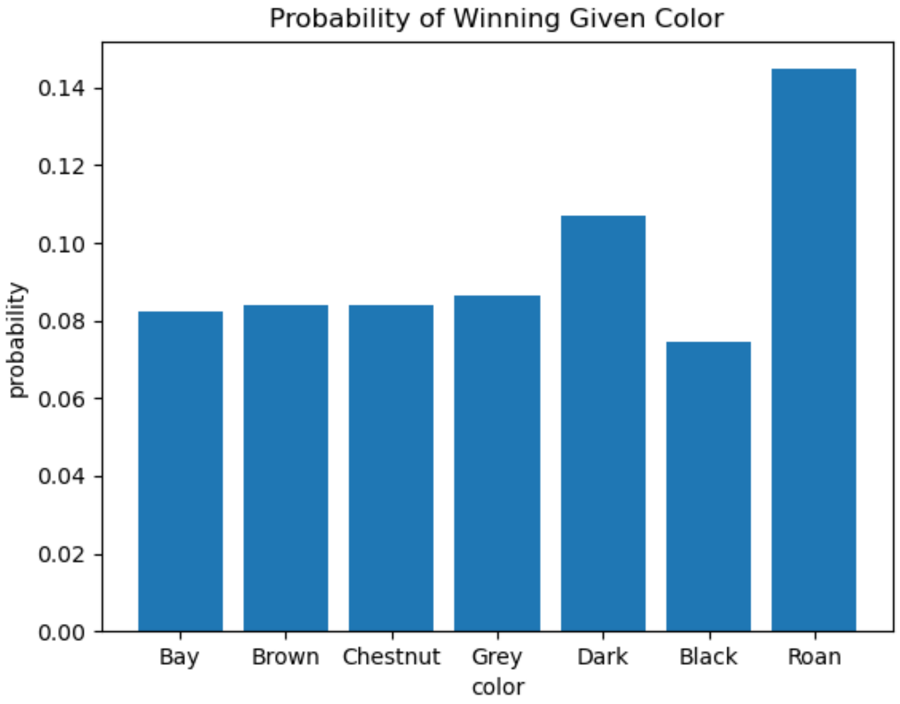




Data Preparation – Feature Analysis (Color)

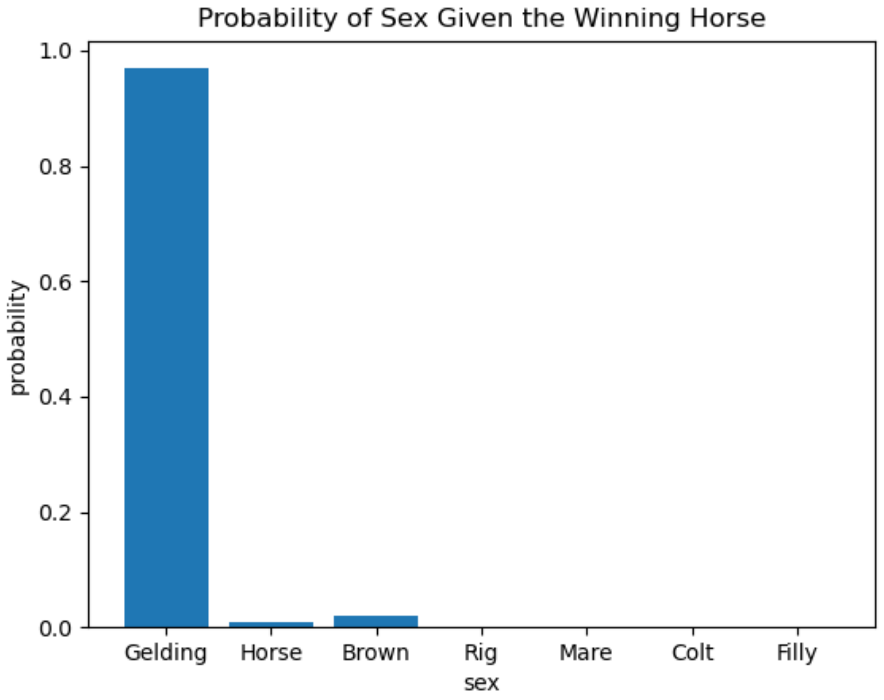
- Most winning horses have skin color Bay

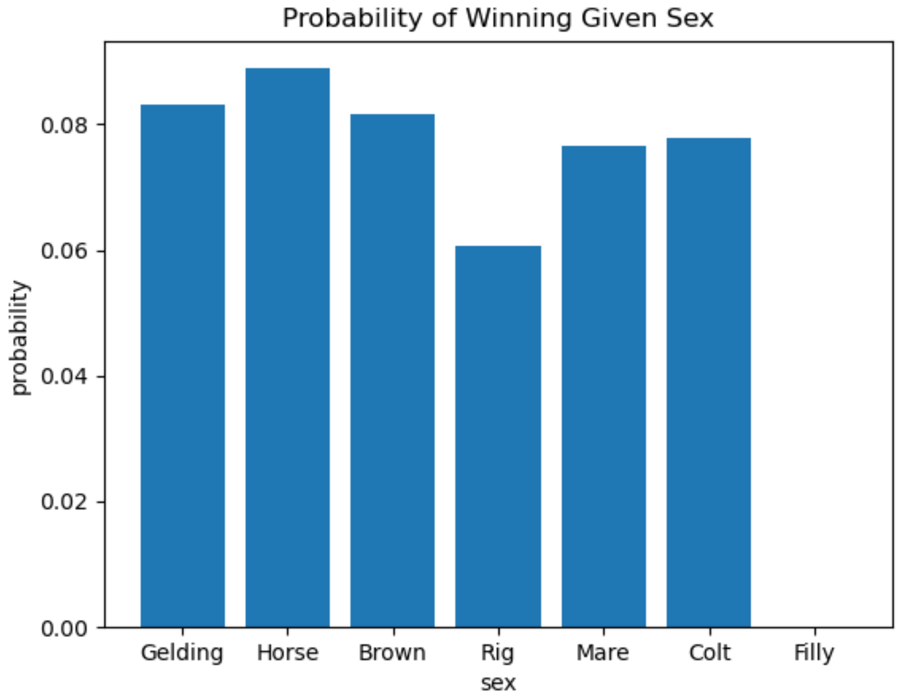




Data Preparation – Feature Analysis (Sex)

- Most winning horses are of sex Gelding.

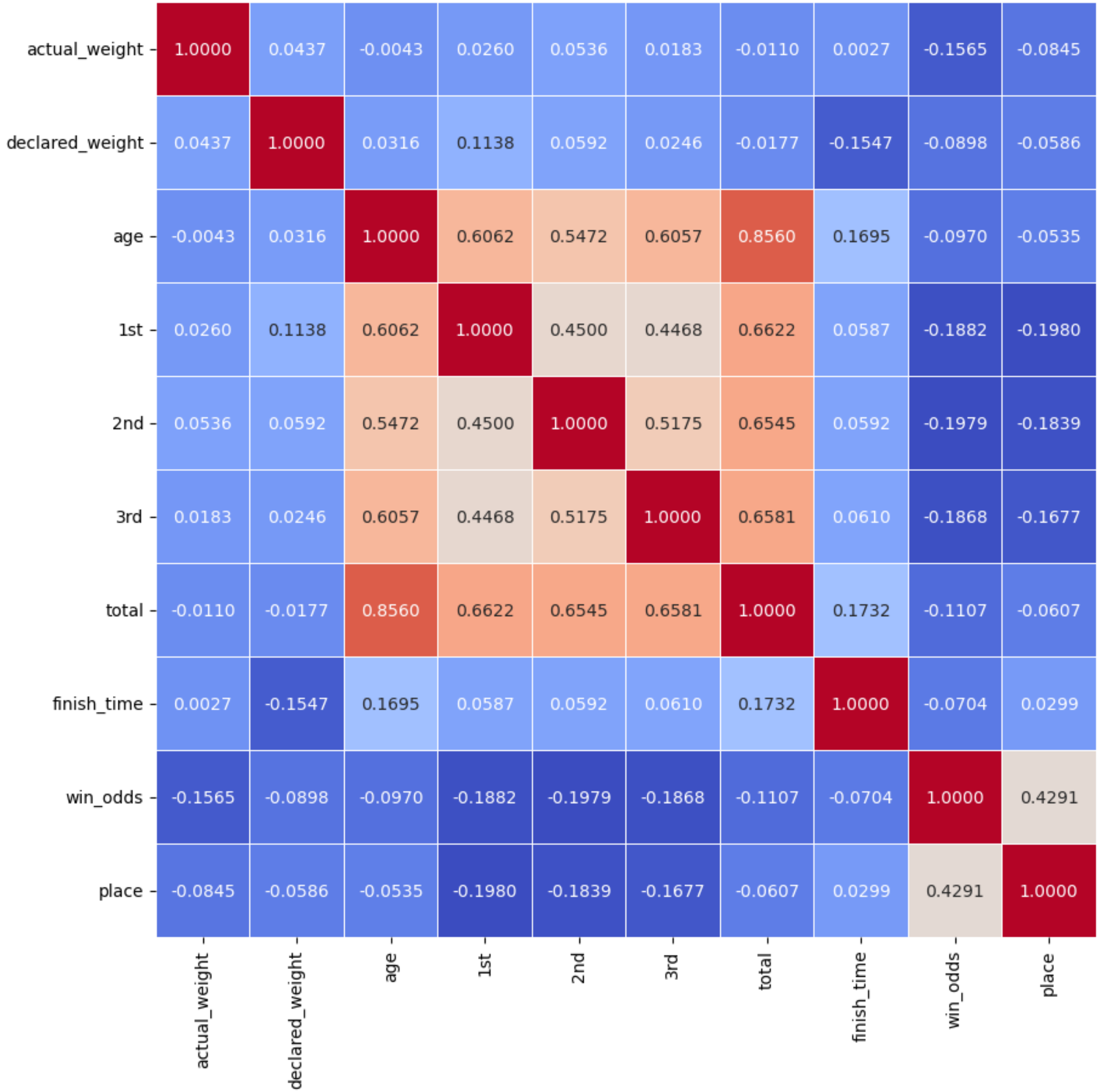




Data Preparation – Feature Analysis (Numerical features)

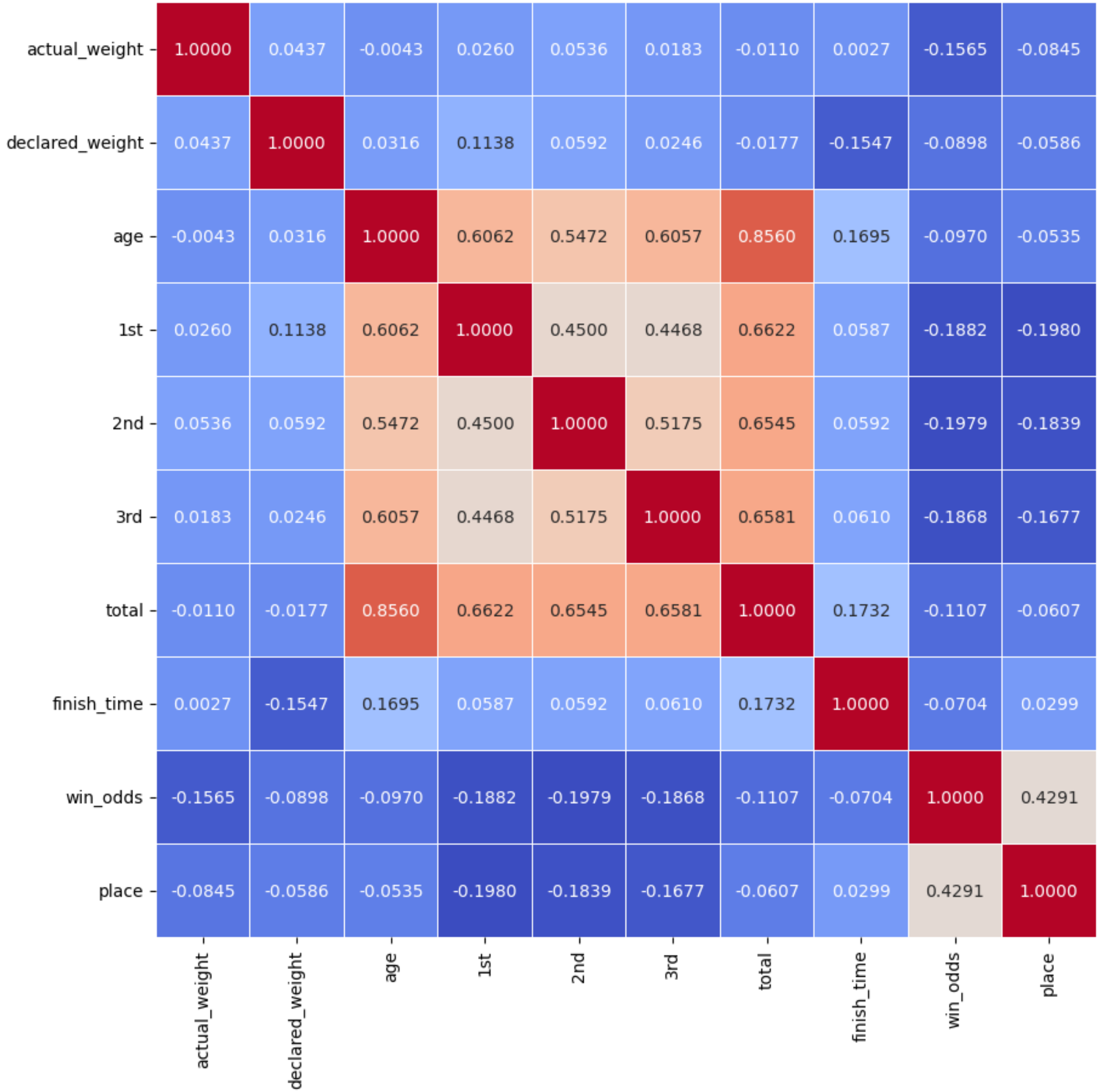
- Frequency of 1st place has a significant correlation with the frequency of 2nd place and 3rd place which are 0.4500 and 0.4468 respectively

- Rating systems are applicable in prediction



Data Preparation – Feature Analysis (Numerical features)

- A positive correlation of 0.4291 between the win odds and the place
- Winning odds help the prediction of horse racing result



Data Preparation – Data Imputation

- A small part of horse data about those retired horses is missing in our data set
- do data imputation on our dataset by using the k nearest neighbors method
- Invoke the KNN Imputer from Scikit Learn library to impute the missing values

Data Preparation – Data Encoding

- Input of our neural network models must be numerical but some of our data are categorical
- One hot encoding[7]
  - dimension of our input will be increased drastically
  - requires extra memory and more computational time in training
- Ordinal Encoding scheme[7]
  - a unique integer means a category
  - dimension of the data is the same as the original
- invoke the Ordinal Encoder from the Scikit Learn Library

Data Preparation – Normalization

- z-score normalization[8]
- Values of all variables are recomputed into the same scale
- the same scale of all variables balances the focus of error minimization in the weight correction algorithm

$$x_i' = \frac{x_i - \bar{x_i}}{\sigma_i}$$

Data Preparation – Rating Generation

- Ratings mentioned before do not exist on the HKJC websites
- need to calculate those ratings with the information provided by our dataset

contest_index	rating_mu	rating_sig	perf_score	place
66	1400	174	1381	9
107	1384	133	1365	11
162	1481	114	1724	5
199	1443	102	1370	9
275	1463	95	1511	7
322	1482	90	1540	7
351	1491	87	1517	6
380	1522	85	1639	6
417	1543	84	1620	4
446	1563	82	1642	6
468	1527	82	1370	10
533	1562	81	1700	2
592	1610	81	1833	1
632	1613	81	1623	5
687	1626	80	1677	6

Methodology - Overview

- Rating systems estimate the relative skill level of horses based on their historical performance
- Self attention mechanism captures the dependencies between horses
- Multiclass classification on place
  - The winning horse number is the output
- Transformer classification model including ratings in the feature list

Methodology - Evaluation

- Accuracy
  - Accurate prediction about the winner
- Betting simulation
  - Net gain
  - Bet $10 for each race in test data

- Combining transformer architecture and ratings give a better result
  - Multilayer perceptron with ratings
  - Transformer without ratings
  - Transformer with ratings

Methodology – Multilayer perceptron

1. Input layer
2. 3 linear hidden layer
3. Dropout layer
4. Output layer

- Relu Activation function
- Cross-Entropy Loss Function
- Stochastic gradient descent

Methodology – Transformer

1. Input layer
2. Word embedding layer
3. Position embedding layer
4. Transformer encoder
5. 2 linear hidden layers
6. Dropout layer
7. Output layer

- Relu Activation function
- Cross-Entropy Loss Function
- Stochastic gradient descent

Experiment and Result – Input Data

- Training data :
  - races from 22 June 2008 to 6 December 2020.
  - 8500 races
- Testing data
  - all races from 9 December 2020 to 17 October 2021.
  - 688 races

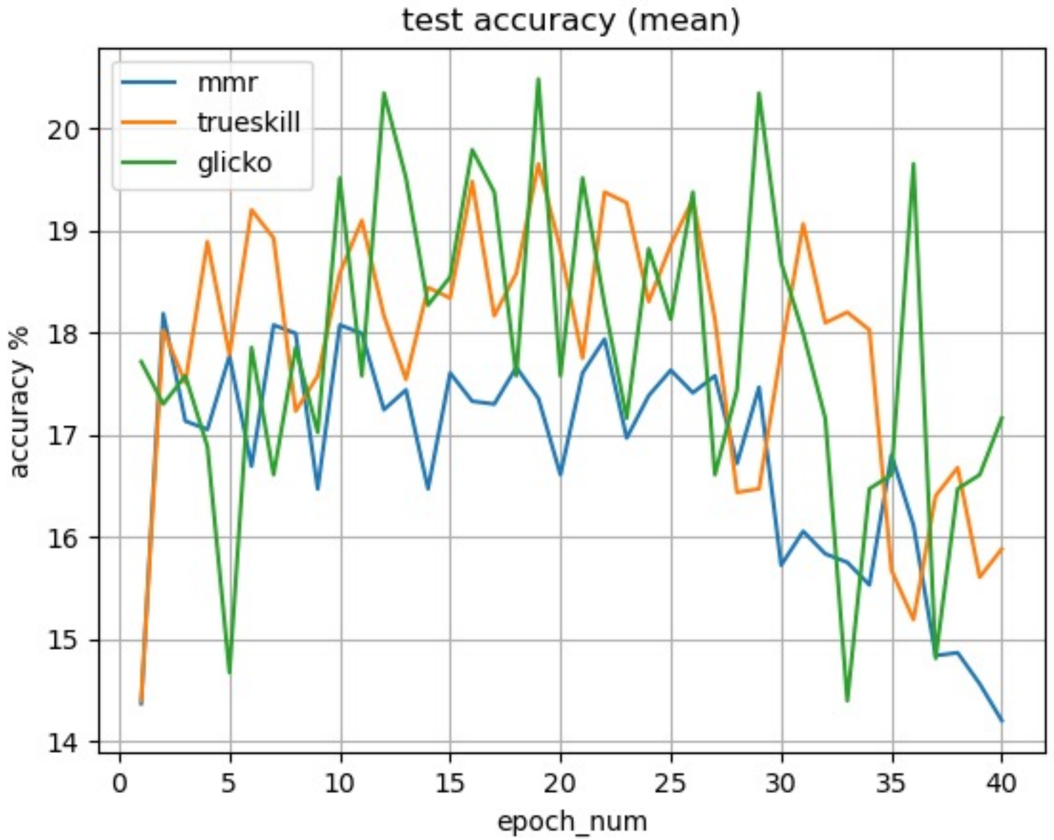
Experiment and Result – Input Data

Feature	Description
Venue	Location of the race
Horse_class	Class of the horses Stronger horses compete in high race class
Distance	The distance of the race
Going	Condition of the lane
Course_track	The lane of the race
Course_track_code	Description about the lane
Horse_i_number	The horse number in the race
Horse_i_name	The name of horse
Horse_i_jockey	The name of jockey
Horse_i_trainer	The name of trainer
Horse_i_declared_weight	The weight of horse
Horse_i_origin	The place of birth
Horse_i_age	The age of horse
Horse_i_color	The color of skin
Horse_i_sex	The gender of horse
Horse_i_1st_place_frequency	The frequency of getting 1st place
Horse_i_total_race	The total count of horse's participation
Horse_i_rating	The rating of the horse

Repeat 14 times

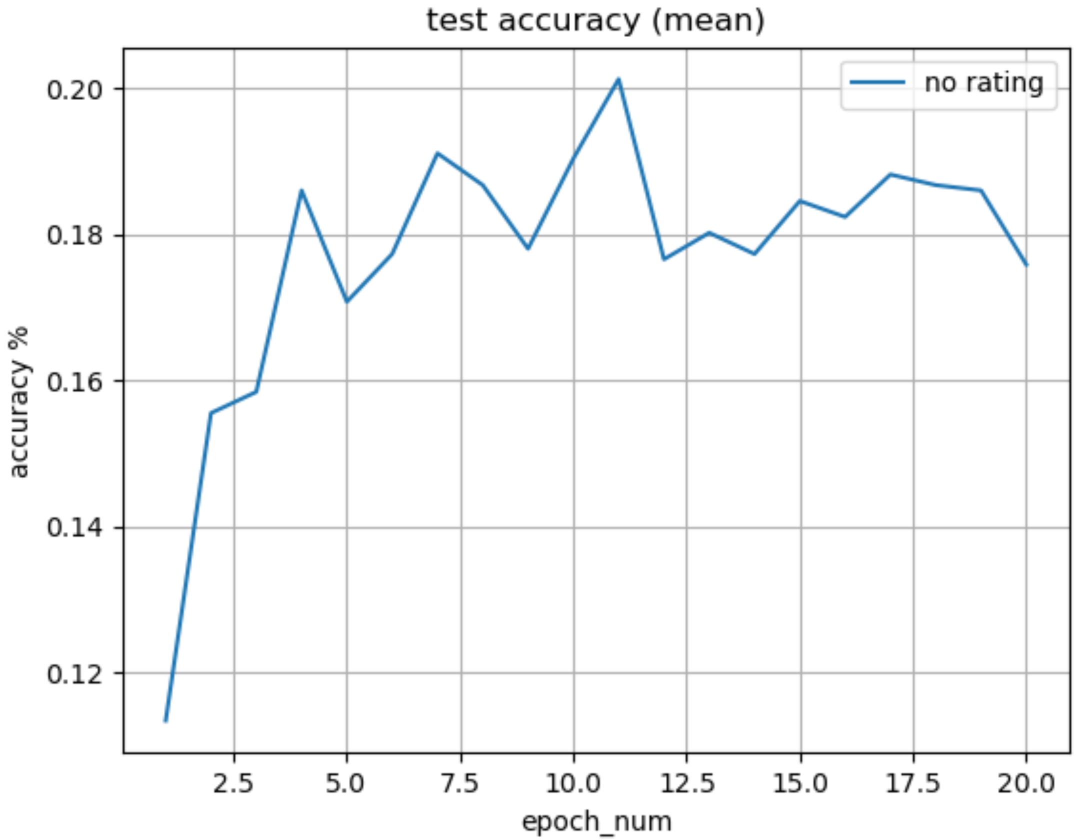
Experiment and Result – Accuracy

- Multilayer perceptron with ratings
- The model with Glicko ratings reaches the highest test accuracy of 20.4%
- The accuracy of the model with Glicko ratings fluctuates in a larger range than that with Elo-MMR and TrueSkill



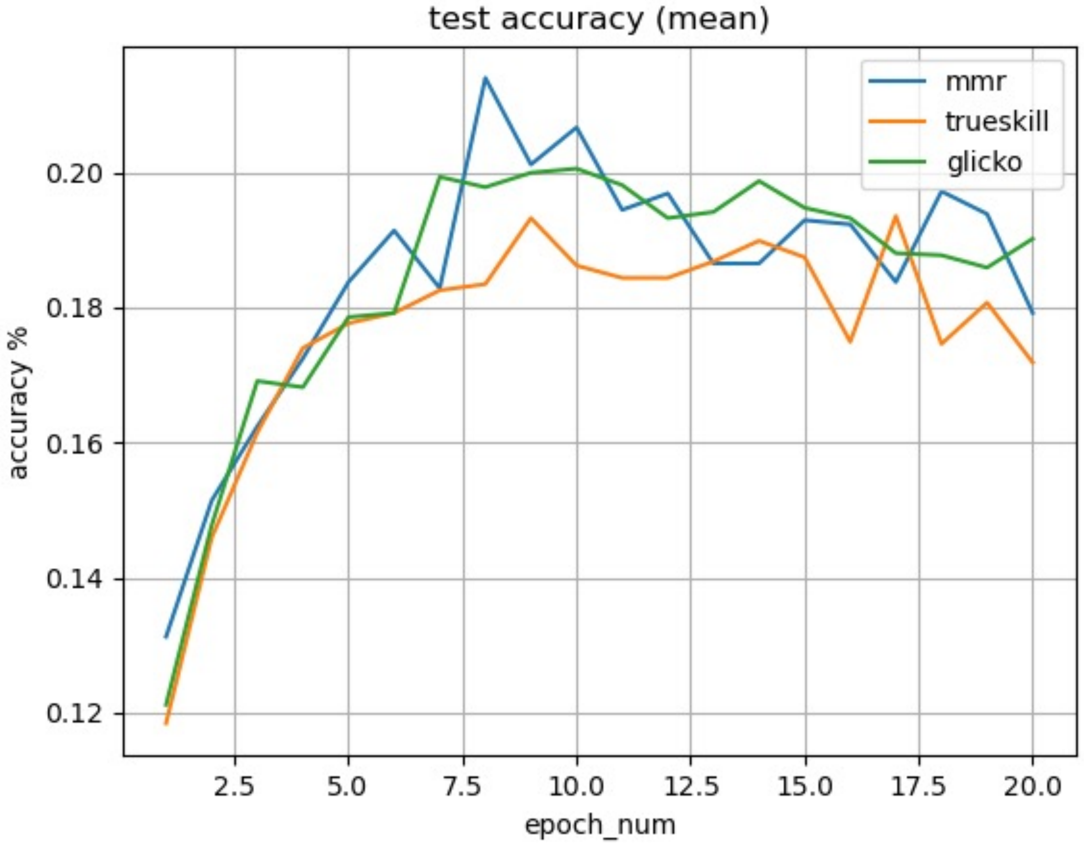
Experiment and Result – Accuracy

- Transformer without ratings
- the best performance of this model is having 19.2% before overfitting



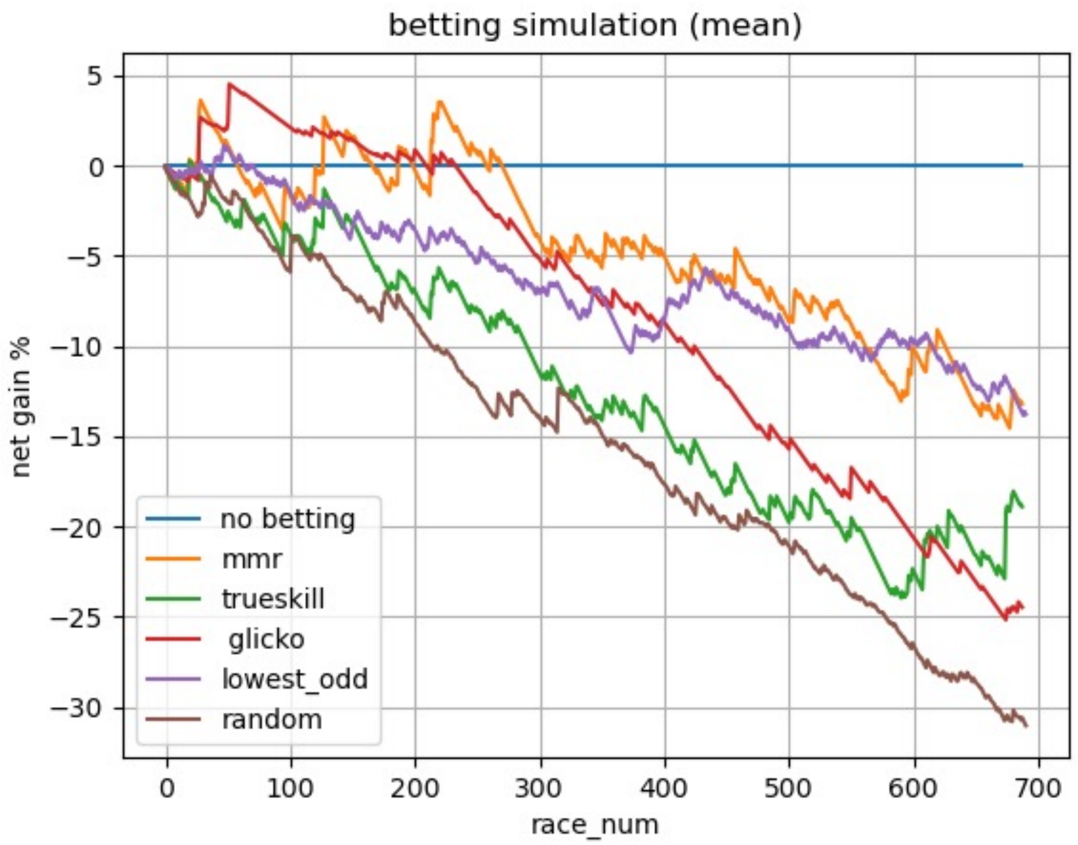
Experiment and Result – Accuracy

- Transformer with ratings
- The transformer model with Elo-MMR ratings has the highest test accuracy of 21.4% among the other models



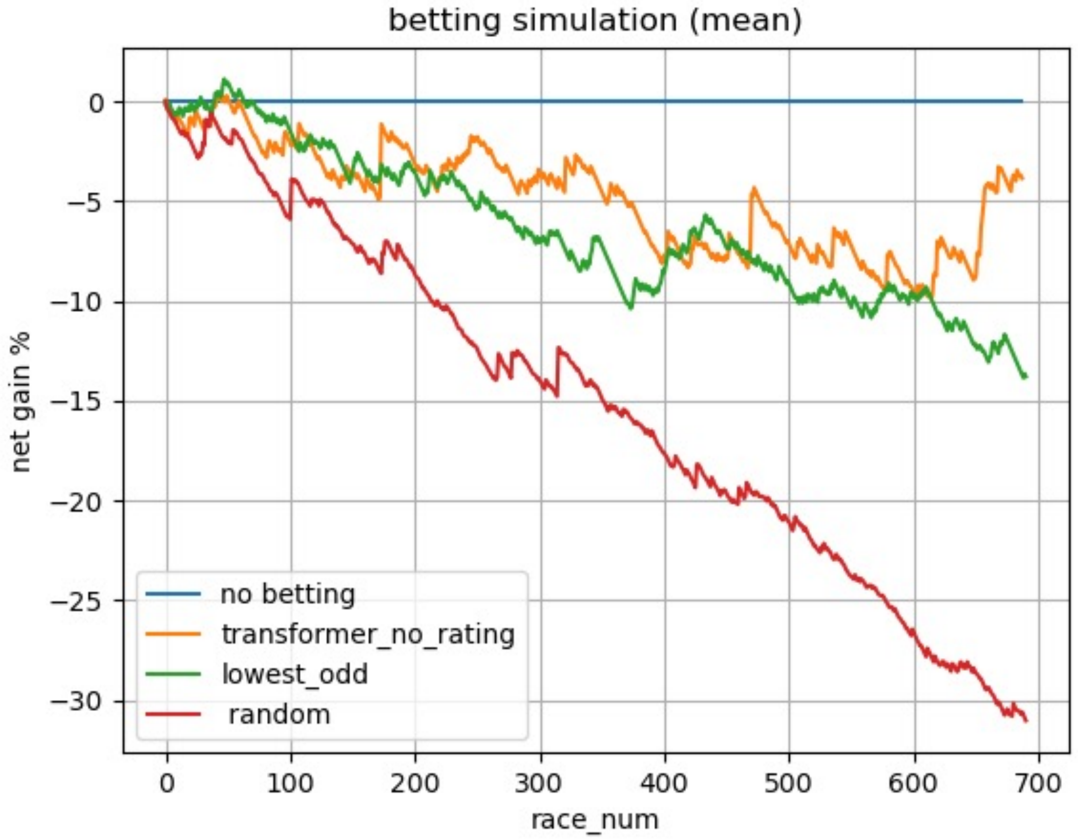
Experiment and Result – Betting Simulation

- Multilayer perceptron with ratings
- all three models perform better than random betting
- Multilayer perceptron with Elo-MMR ratings has the best performance
  - highest net gain : -13%



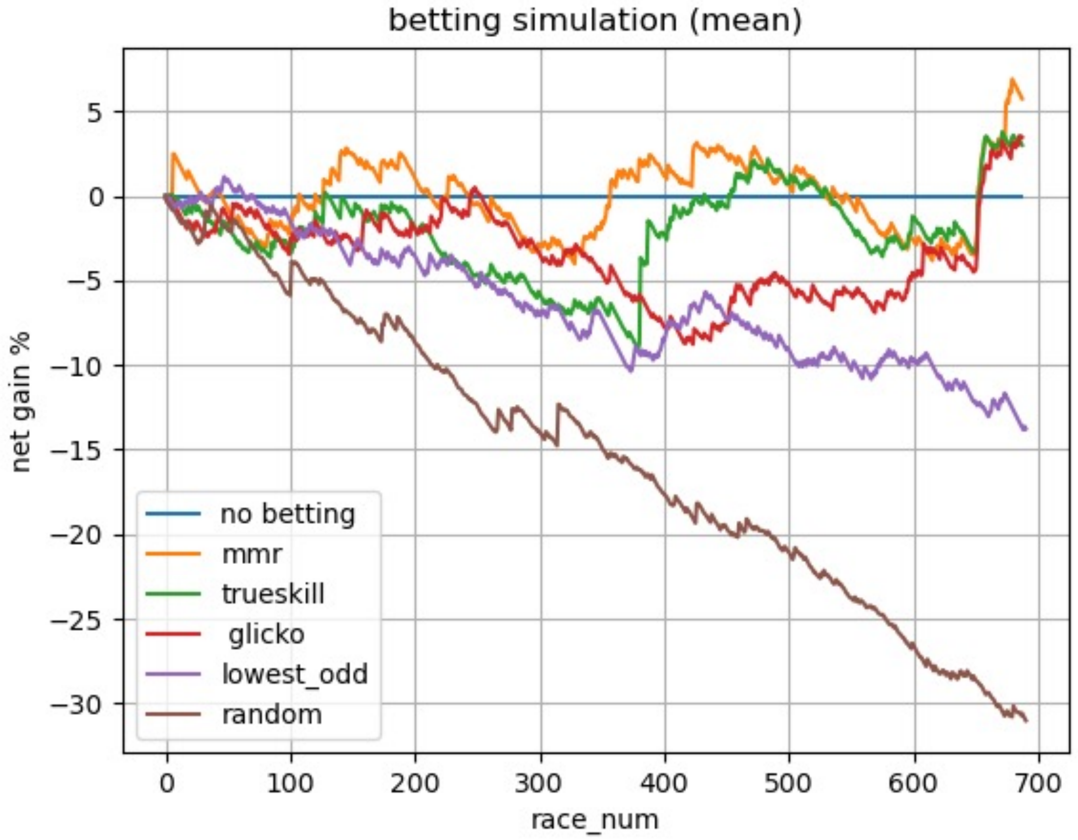
Experiment and Result – Betting Simulation

- Transformer without ratings
- The net gain is -4% after betting on all 688 races in our test data
- Better than the multilayer perceptron



Experiment and Result – Betting Simulation

- Transformer with ratings
- Positive net gain of 3% to 6% after betting on 688 races in the test data
- Transformer with Elo-MMR ratings has the best performance



Conclusion

- Win odds in the feature list have the effect of enhancing the accuracy and net gain
- Exclude the win odds from the feature list this time
- Resemble the effect of the winning odds by combining rating systems and the transformer architecture
- the best case of our models is the transformer with Elo-MMR ratings
  - the highest test accuracy : 21.4%
  - a positive net gain : 6%

Thank you!

References

[1] Y. Liu and Z. Wang, "Predicting Horse Racing Result with Machine Learning," Department of Computer Science and Engineering, Hong Kong, 2018.

[2] Y. Wong, "Horse Racing Prediction using Deep Probabilistic Programming with Python and PyTorch (Uber Pyro)," Department of Computer Science and Engineering, 2018.

[3] GLICKO (Glickman, Mark E. "The glicko system." Boston University 16 (1995): 16-17)

[4] Herbrich, Ralf, Tom Minka, and Thore Graepel. "Trueskill™: A Bayesian skill rating system." Proceedings of the 19th international conference on neural information processing systems. 2006.

[5] Ebtekar and P. Liu, “Elo-MMR: A rating system for massive multiplayer competitions,” Proceedings of the Web Conference 2021, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, et al., "Attention Is All You Need", CoRR, vol. abs/1706.03762, 2017.

[7] K. Potdar, T. S., and C. D., “A comparative study of categorical variable encoding techniques for neural network classifiers,” International Journal of Computer Applications, vol. 175, no. 4, pp. 7–9, 2017.

References

[8] J. Sola and J. Sevilla, “Importance of input data normalization for the application of neural networks to complex industrial problems,” IEEE Transactions on Nuclear Science, vol. 44, no. 3, pp. 1464–1468, 1997.