

The Chinese University of Hong Kong

Final Year Project Report (Term 1)

Betting Odds Calculation with Machine Learning

Author:

NAM Man Leung

Supervisor:

Prof. LYU Rung Tsong Michael

LYU2102

Faculty of Engineering

Department of Computer Science and Engineering

30 November 2021

The Chinese University of Hong Kong

Abstract

Faculty of Engineering

Department of Computer Science and Engineering

BSc degree in Computer Science

Betting Odds Calculation with Machine Learning

by Nam Man Leung

Winning odds in horse racing reflects the public opinions because the more confidence the public about the winning of a horse, the lower the winning odds due to the pari-mutuel betting system. The transformer model in natural language process has shown a success in dealing with prediction of sequence input but there is not any research exploring the use of transformer model in horse racing prediction. The ratings given by rating systems have been used in many competitions to represent the skill level players. In this project, we combine these two techniques for horse racing prediction and see if it can have the same effect as the winning odds in helping the prediction. By comparing the results of using each technique alone, we show that the combination of the two technique can achieve better prediction accuracy, positive net gain and less training time.

Acknowledgement

I would like to express my gratitude to my supervisor Professor Michael R. LYU and my advisor Mr. HUANG who provided me suggestions and guidance throughout the project.

I would also like to show appreciation to previous students' work in horse racing prediction which strengthen my background knowledge in doing this project.

Table of Contents

Chapter 1	7
Overview	7
1.1 Introduction.....	7
1.2 Background	10
1.2.1 Horse Racing in Hong Kong.....	10
1.2.2 The Hong Kong Jockey Club.....	10
1.2.3 Pari-mutuel betting in horse racing	11
1.2.4 Types of bet.....	11
1.3 Motivation.....	14
1.4 Objective	15
1.5 Related Works	16
Chapter2.....	19
Preliminary.....	19
2.1 Background knowledge of Rating system	19
2.1.1 Glicko Rating System	19
2.1.2 TrueSkill Rating system.....	22
2.1.3 Elo-MMR rating system	26
2.2 Background knowledge of Transformer	29
2.2.1 Transformer.....	29
2.2.1.2 Model architecture	30
2.2.1.2 Encoder	31
2.2.1.3 Decoder.....	31
2.2.1.4 Attention.....	32
2.2.1.5 Scaled Dot-Product Attention	32
2.2.1.6 Multi-Head Attention.....	33
2.2.1.7 Positional Encoding	34
2.3 Evaluation Strategy.....	35
2.3.1 Random betting (Profit-making aspect).....	35
2.3.2 Lowest odd betting (Profit-making aspect).....	35
2.3.3 Multilayer perceptron prediction (accuracy aspect)	35
2.3.4 Transformer without rating in the input (accuracy aspect)	36
Chapter 3.....	37
Data Preparation.....	37
3.1 Data Collection	37
3.2 Data Description	37
3.2.1 Racing Record.....	38

3.2.2 Horse Information.....	39
3.3 Data Analysis	40
3.3.1 Categorical Features.....	41
3.3.1.1 Age	41
3.3.1.2 Origin	43
3.3.1.3 Color	45
3.3.1.4 Sex.....	47
3.3.1.5 Draw.....	49
3.3.2 Numerical Features	51
3.3.2.1 Frequency of 1 st Place.....	52
3.3.2.2 Finish time	52
3.3.2.3 Win odds	53
3.4 Data Preprocess.....	54
3.4.1 Data Imputation	54
3.4.2 Data Encoding.....	55
3.4.3 Normalization	56
3.4.4 Rating Generation	57
Chapter 4.....	59
Methodology	59
4.1 Methodology Overview	59
4.2 Model Design.....	62
4.2.1.1 Multilayer Perceptron Classification	62
4.2.1.2 Multilayer Perceptron Architecture.....	62
4.2.2.1 Transformer Classification.....	64
4.2.2.2 Transformer Architecture	64
Chapter 5.....	66
Experiment and Result.....	66
5.1 Input Data.....	66
5.2 Results.....	68
5.2.1 multilayer perceptron classification model	68
5.2.1.1 Accuracy	68
5.2.1.2 Betting simulation.....	70
5.2.2 Transformer classification model without ratings.....	71
5.2.2.1 Accuracy	71
5.2.2.2 Betting simulation.....	73
5.2.3 Transformer classification model with ratings.....	74
5.2.3.1 Accuracy	74
5.2.3.2 Betting simulation.....	76

Chapter 6.....	77
Conclusion and Future work.....	77
6.1 Conclusion	77
6.2 Future Work	78
References.....	79

Chapter 1

Overview

Understanding the win odds in horse racing with machine learning methods is the purpose of this final year project. As the win odds is related to the inverse of winning probability of horses, we first attempt to do the prediction of the winning horse and use the result of the prediction for win odds calculation in the next stage. The introduction to machine learning methods and the background about horse racing are provided in the beginning of this section. Then, our motivation towards this project and the respective objectives of the first semester and the second semester will be stated.

1.1 Introduction

Machine learning has become a hot topic in technical fields with the dramatic advancement of the hardware and appearance of big data in recent years. It has been applied to solve different real-world problems such as weather forecast, image recognition, speech recognition and natural language process etc. The concept of machine learning is optimizing the parameters defined in a model with the guidance of training experience to get intuition and prediction [1]. Machine learning is not a specific to one particular field, but the junctions of different domains such as statistics, computer science and data science. For instance, it uses the knowledge in statistics to build models and knowledge in computer science to convert the models into computer's representations and design an efficient algorithm to deal with the optimization problem of the model [2].

Machine learning can be divided into two types. The first type is supervised learning in which the known target outputs are used to correct the values of the parameters in the mapping model between the input and output [3]. The mapping will then be employed for predicting the output of new incoming data. The second type is unsupervised learning in which there are no explicit target output to guide the optimization of parameters in model. Instead, an assessment of the representation's quality is learned in a self-organizing process [4]. For this project, supervised learning is our choice because the win odds can easily be collected from the HKJC website.

The primitive neural network architecture in machine learning was the single layer perceptron proposed in 1958. It was further developed into a multilayer perceptron in 1975 for solving nonlinear problems and linearly separable problems that cannot be solved by the perceptron [5]. The multilayer perceptron gradually evolves to different kinds of neural network architecture such as deep neural network, convolutional neural network, recurrent neural network and long/short term memory network. The original design of neural network was to emulate how the brain function in doing a task by treating each neuron in the neural network as the neuron in the brain and aggregating them into a complicated information system which is nonlinear in nature [6].

Natural Language processing has been the popular topic in the research field, and it had initially addressed by the convolutional neural network and recurrent network due to their exceptional performances until the appearance of the transformer architecture in 2017 which has an even better performance in understanding and generating the natural language by parallel training and ability to tackling lengthy sequence inputs [7].

Several attempts in horse racing prediction with machine learning methods were done by previous FYP students. LYU1603 tried to predict the winning horse with regression on time [8]. LYU1703 tried to predict the winning horse and the places with MLP and rank network [9]. LYU1805 tried to predict the winning horse with deep probabilistic programming [10]. For this project, we approach the horse racing prediction from a different perspective. Since both the inputs of this horse racing prediction and natural language processing are sequences, we decide to reduce the horse racing prediction to a natural language processing classification problem. We hope that the techniques in natural language processing can capture the relationships between horses in a single race and do the prediction according to the dependency. There are three contributions of this project. The first contribution is applying the transformer model in horse racing prediction which has not been explored yet. The second contribution is discovering that training a transformer model in horse racing context requires fewer epochs than using traditional neural network model [8][9][10]. The third contribution is showing a positive net gain when using the prediction of transformer model with ratings as input in horse racing betting.

1.2 Background

1.2.1 Horse Racing in Hong Kong

Horse racing in Hong Kong is a sport competition introduced from British which usually has 10 – 14 jockeys riding on corresponding horses in a single race competing to reach the finish line in a shorter time. It has been an esteemed sport event in Hong Kong for over 100 years as betting is allowed for people to bet on the horses which they like. This event is mostly held on Sundays and Wednesdays. There are total 10 day races on Sundays and 8 night races on Wednesdays respectively. The number of competitors is limited to 14 for races on Sundays while it is limited to 12 for races on Wednesdays. In each year, the horse racing season starts in September and ends in July and there are roughly 88 days having the horse racing within a season [11].

1.2.2 The Hong Kong Jockey Club

The Hong Kong Jockey Club, founded in 1884, is a certified non-profit making and charitable organization responsible for hosting horse racing events and other betting entertainments. It gains enormous revenue from its sport betting events every year and those revenue will be split for operational costs and return to the community. HK\$29.4 billion was returned to the community in terms of duty, tax and donations in 2020-2021 [12].

1.2.3 Pari-mutuel betting in horse racing.

In pari-mutuel betting, the bets from people are accumulated to a pool in each race. The bookmaker will take a fixed percentage from the pool [13]. In Hon Kong, The Hong Kong Jockey Club acquires 17.5% of the pool in winning bets as its revenue and allocates the remaining in the pool to the betters with a correct prediction with reference to the odd which is the ratio of return to the bet calculated before the start of the race. The odd cannot be interpreted as the true winning probability of a horse, but it is just an estimation of how many betters who favors the horse. In other words, it reflects the public intelligence. Since the betters are indeed betting against each other, positive net gain is expected if we do prediction that is more accurate than the public [14].

1.2.4 Types of bet

The Hong Kong Jockey Club provides various types of bets for bettors. The types and explanation can be found in Figure 1.

		Telephone Betting	Interactive Services [#]	Off-Course Betting Branches		Racecourses	
				Self Vending Terminal	Service Counter	Self Vending Terminal	Service Counter
Minimum Investment Amount	Horse Racing	Pari-Mutuel Pools					
		From 30 minutes before Race 1 up to the last race starts, <ul style="list-style-type: none">• HK\$20 until 10 minutes before the start of each race• HK\$50 during the last 10 minutes of each race	Applicable IS Device(s) HK\$10(any time)	HK\$10 (any time)	HK\$20 (30 mins before Race 1 to start of last race)	HK\$10 (any time)	HK\$10 (any time)

Figure 1. Pari-mutuel betting provided by the HKJC [15]

As we see from Figure 1, the minimum amounts to invest in the pari-mutuel pools is \$10 from self-vending terminal such as the HKJC mobile application or the HKJC WEB Application at any time.

Single-race Pool	Dividend Qualification
Win	1st in a race
Place	1st, 2nd or 3rd in a race, or 1st or 2nd in a race of 4 to 6 declared starters (applicable to local races) 1st, 2nd, 3rd or 4th in a race, or 1st, 2nd or 3rd in a race of 7 to 20 declared starters, or 1st or 2nd in a race of 4 to 6 declared starters (applicable to designated simulcast races)
Quinella	1st and 2nd in any order in a race
Quinella Place	Any two of the first three placed horses in any order in a race
3 Pick 1 (Composite Win) Winning Trainer (Composite Win) Winning Region (Composite Win)	Composite containing the 1st horse in a race
Forecast	1st and 2nd in correct order in a race
Trio	1st, 2nd and 3rd in any order in a race
Tierce	1st , 2nd and 3rd in correct order in a race
First 4	1st, 2nd , 3rd and 4th in any order in a race
Quartet	1st, 2nd , 3rd and 4th in correct order in a race

Table 1. Types of bets in the single race pool [16]

The single race pool and the dividend qualification for beginners are shown in Table 1.

Multi-race Pool	Dividend Qualification
Double	1st in each of the two nominated races Consolation : 1st in 1st nominated race and 2nd in 2nd nominated race
Treble	1st in each of the three nominated races Consolation : 1st in the first two Legs and 2nd in 3rd Leg of the three nominated races

Table 2. Type of bets in multi-race pool [16]

The multi-race pool and the dividend qualification for more experienced bettors are shown in Table 2

1.3 Motivation

Horse racing held by the Hong Kong Jockey Club has been the most favored sport betting event in Hong Kong and its popularity can be shown by the colossal amount of revenue which is approximately HK\$280 billion in 2020-2021 despite economic downturn caused by the coronavirus pandemic [17].

Tremendous efforts have been made to predict the winning horse of each race by machine learning, but the outcome has yet been unsatisfied as profitable results can only be attained under certain circumstances. It is believed that the betting odds hide the secret of profitable plans from the observation which bookmakers are consistently having interests by providing profitable betting odds to gamblers. Therefore, investigating the betting odds calculation may help reveal the reasons of the bookmaker's enormous financial gain.

1.4 Objective

The objective in this project is to reproduce the effect of winning odds from the Hong Kong Jockey Club in horse racing prediction. As the horse with a low winning odd usually has a higher winning probability as implied from the public intelligence, we partition the overall project objective into two objectives which are winning horse prediction and winning odds calculation from the result of winning horse prediction. These objectives are planned to be achieved in two semesters.

First Term:

- Convert the data collected from the HKJC into a sequence that can be fitted to a natural language processing model.
- Find other features that have similar meaning as the winning odds
- Build a natural language process model for winning horse classification
- Evaluate the performance of the proposed model on the test set.

Second Term:

- Improve and modify the class classification model in the first term to a multiclass classification model for predicting the places of horses
- Calculate the winning odds of horses with the places predicted from the multiclass classification model and other features by a machine learning model.

1.5 Related Works

Researchers have been interested in applying machine learning methods in learn the complex relationship in sport betting and predicting the outcome accurately. Several studies investigated horse racing prediction by artificial neural network [18], conditional logistic regression [14], random forest [20] and support vector machine [21].

Elnaz and Khanteymooori [18] applied artificial neural network in horse racing prediction with five different supervised neural network learning algorithms which are Conjugate Gradient Descent, Quasi-Newton, Levenberg-Marquardt, Backward-Propagation and Backward-Propagation with Momentum. The experiment used the horse racing records in January 2010 in the United States and the result was exceptional that all learning algorithms produced satisfying predictions of 77% accuracy in average. The performance differences between the learning algorithms are small. Although Backward-Propagation took a longer training time, it achieved a slightly better prediction result than others. Overall, this research demonstrated that artificial neural network was applicable to horse racing prediction.

Silverman and Suchard [14] proposed adjustments to multinomial logit model for horse racing prediction which was suggested by Bolton and Chapman [19]. They exploited the winning dividends by introducing a frailty contribution and parameter regularization to the regression model. They collected the data of 3681 races in Hong Kong from the HKJC and 737 races were retained for testing the model. They discovered that they could gain a remarkable higher return by changing the objective to simply increasing the profit and combining a calculated inverse-frailty score in the

in the experiment.

Lessmann, Sung and Johnson [20] explored alternative methods in predicting horse racing results. They admitted that the conditional logit model was a proper tool for estimating the winning probability of a horse in conjunction with other horses in a race. In addition to that, they showed that random forest could complement the conditional logit-based horseracing forecasting. Consequently, they adapted a two-stage modelling framework which captured the complicated relationship between horse's information and the results of races in the first stage. Then, the winning probability of a horse within a single race was computed at the second stage. In the second stage, random forest was used in revealing the winner horse by counting the number of votes regarding whether the horse was a winner from the decorrelated decision trees.

Chung, Change and Ko [21] utilized the support vector machine in the prediction of horse racing results in Hong Kong. They divided their training data into multiple similar training sets and train a support vector machine for each training set. For those weaker models, they were combined to form a stronger model. The outcome of a race was determined in a similar way as random forest. All trained support vector machines formed a committee machine and did voting. In the experiment, they collected data from the HKJC official website. There were 33532 horse records and 2691 race records dated from 1st Jan 2012 to 30th June 2015 in the dataset. The result of the experiment showed a 70.86% accuracy in predicting the winner horse by the committee machine.

Tung and Hei [8] attempted to build a classification model for winning horse prediction with Tensorflow. They used the neural network to build a binary

classification model and betted on the horse with a if the prediction of the model revealed that the horse was a winner. They set a confident threshold to be 0.8 so that they only betted the horse when the model predicted it as a winner with confident threshold exceeding 0.8. As a result, they exhibited a 30% net gain after one year.

Liu [9] tackled the horse racing prediction problem by building a supervised neural network in predicting the finishing time of each horse. After that, he did comparison between horses' predicted finishing time and ranked the horses based on it. He set a confident threshold to be 0.5 and betted only on class 1 and class 2 races. This setting was shown to have a positive net gain over a full race season.

Wong [10] applied Pyro which was a probabilistic programming language supported by Python for building sophisticated probabilistic models. Automatic differentiation, neural networks and backward propagation were assisted by the PyTorch backend. The abstraction provided by the probabilistic programming language simplify the code for inferences and probabilistic sampling. The result of the experiment showed a profit of 14.43% could be gained when using features including the winning odds while it dropped to 7.59% when using features excluding the winning odds.

Chapter2

Preliminary

2.1 Background knowledge of Rating system

2.1.1 Glicko Rating System

Glicko rating system [22] is an extension to the Elo rating system. It is a statistical model that addresses the limitation of the Elo rating system by introducing an additional measurement the rating deviation. This measurement is intended for assessing the reliability of a player's rating. When the value of rating deviation is high, it infers that the player has not played the game for a long period of time and the rating thus becomes unreliable. In contrast, a low value of rating deviation indicates that the player plays the game frequently and the rating is more reliable. The intuition is that the uncertainty of a player's ability reduces because more information is obtained by the player plays more games.

The rating and the rating deviation of horses are calculated in two steps. The formula is recursive in nature as the result of the current rating and rating deviation are determined from the rating and rating deviation from the last rating and last rating deviation.

At the new rating period, we should compute the rating and rating deviation for each horse based on its previous rating and rating deviation. In step 1, we focus on the rating deviation.

If the horse is new to the race which means it hasn't participated in any races, we assign 1500 and 350 to its rating and rating deviation respectively. Both 1500 and 350 are default values of the rating and the rating deviation.

If the horse participated races in the past, we take its rating from the last race for computing the current rating deviation with the formula below,

$$\sigma = \min(\sqrt{\sigma_{old}^2 + c^2}, 350). \quad (1)$$

σ is the current rating deviation and σ_{old} is the rating deviation of the last race. c is the constant controlling the uncertainty between races. The current rating deviation is the minimum value between the computation from the old rating deviation and 350.

In step 2, we do the update of the rating and rating deviation for each horse in a race. Let r be the rating of a horse in the last race and σ be the rating deviation computed in step 1. Then, r_1, r_2, \dots, r_n are the rating of the other horses from their last rating period. The corresponding rating deviation is $\sigma_1, \sigma_2, \dots, \sigma_n$. The result of horses in the race is s_1, s_2, \dots, s_n . If the horse win in the race, s_i equals to one. If the horse loses the race, s_i equals to zero.

Let r_{new} and σ_{new} be the updated rating and rating deviation of a horse and we repeat this procedure for each horse.

We first define the following terms,

$$q = \frac{\ln(10)}{400} \quad (2)$$

$$g(\sigma) = \frac{1}{\sqrt{1 + \frac{3q^2(\sigma^2)}{\pi^2}}} \quad (3)$$

$$E(s|r, r_j, \sigma_j) = \frac{1}{1 + 10^{-g(\sigma_j)(r - r_j)/400}} \quad (4)$$

$$d^2 = (q^2 \sum_{j=1}^n (g(\sigma_j))^2 E(s|r, r_j, \sigma_j) (1 - E(s|r, r_j, \sigma_j)))^{-1} \quad (5)$$

The above terms are used in the update.

$$r_{new} = r + \frac{q}{1/\sigma^2 + 1/d^2} \sum_{j=1}^n g(\sigma_j) (s_j - E(s|r, r_j, \sigma_j)) \quad (6)$$

$$\sigma_{new} = \sqrt{\left(\frac{1}{\sigma^2} + \frac{1}{d^2}\right)^{-1}}. \quad (7)$$

2.1.2 TrueSkill Rating system

TrueSkill rating system [23] also measure the uncertainty of player skill level, but it also has additional features to the Glicko rating system. The first one is the relaxation to the number of players in a game. As Glicko rating system is designed for 2-players chess games, it assumes that there are only one winner and one loser in each game.

The TrueSkill rating system tries to adapt to a multiple player environment by assuming that the outcome of each game is a permutation of multiple teams or players so that it is dedicated for multiplayer games. The second one is the inference for individual skills in games which requires players to form teams. In our situation, each team only has one player because horses in horse racing does not form a team and we treat each horse as a team.

We apply the Trueskill rating system in horse racing in which there are n horses $\{1, \dots, n\}$ in a race and each individual horse form a team with only one member. Let $T := \{T_1, \dots, T_n\}$ and T_i be the i -th team which has horse i as the only team member so that $T_i \cap T_j = \emptyset$ for $i \neq j$. We also let $R := (r_1, \dots, r_n)$ be the result of each team in a race. If the i -th horse wins in a race, then $r_i = 1$. Otherwise, $r_i = i$ if the i -th horse gets the i -th place in the race.

As our goal is estimating the skill level of horses, we would like to calculate the probability that the players have skill level S given the result of the race R and the team assignment T . From the training dataset, we have the race result given the team assignment T and skill level S . Therefore, we can obtain the probability $P(R|S, T)$ of the race with R as the race result and S as the skill level horse all participating horses.

Then, $P(S| R, T)$ can be obtained by Bayes' rule,

$$P(S| R, T) = \frac{P(R| S, T) P(S)}{P(R | T)}. \quad (8)$$

We assume the skill level of each horse is a Gaussian distribution with parameters μ_i and σ_i so that $P(S) = \prod_{i=1}^n N(s_i; \mu_i, \sigma_i)$. The race performance of each team T_i is actually the race performance of each horse because every team has only one horse as the member. So, the race performance t_i of T_i is modelled as $N(p_i; s_i, \beta^2)$. We then order the teams in ascending order based on its rank so that the order of team is $r_{(1)} \leq r_{(2)} \leq \dots \leq r_{(n)}$. As a result, the probability that the race has outcome R given the teams T is the following,

$$\begin{aligned} P(R | T) &= P(R | \{T_1, \dots, T_n\}) \\ &= P(t_1 > t_2 > \dots > t_n). \end{aligned} \quad (9)$$

Assume a very simple horse race with 3 teams and each team has only one horse so that $T_1 = \{1\}$, $T_2 = \{2\}$ and $T_3 = \{3\}$. Also, team 1 is the winner while team 2 gets the second place and team 3 gets the third place respectively. The joint distribution $P(S, t | R, T)$ can be represented by the factor graph below.

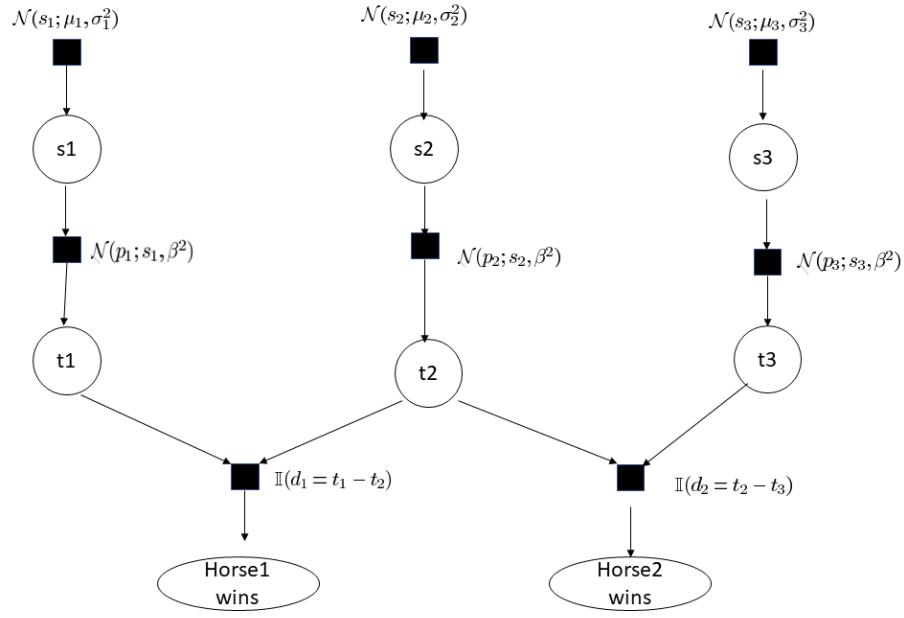


Figure 2. The factor graph describing the joint distribution

In Figure 2, the gray circles indicate the variables, and the black squares indicate the factor nodes respectively. The joint distribution $P(S, t | R, T)$ is computed by the product of all the functions next to the factor nodes. The dependent relationships of the factors are reflected from the graph and the graph structure is utilized for an efficient inference algorithm.

As we have the joint distribution from the factor graph, we can get back the posterior distribution of skill level of horses given R and T $P(S | R, T)$ by integrating the team performances t_i which is the same as the individual horse performances,

$$P(S | R, T) = \int_{-\infty}^{\infty} P(S, t | R, T) dt. \quad (10)$$

In the factor graph, the results at the bottom will be used for update in the approximate message passing part and the update equations for each section are shown in the Figure 3.

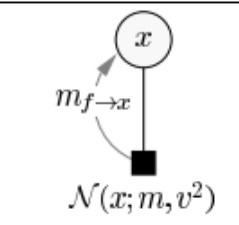
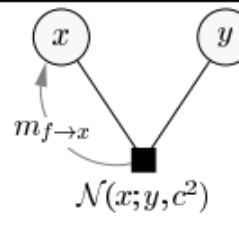
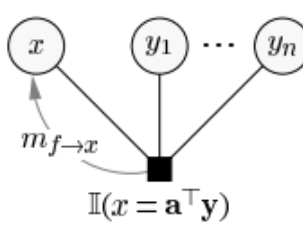
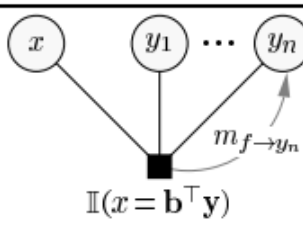
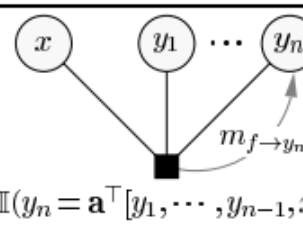
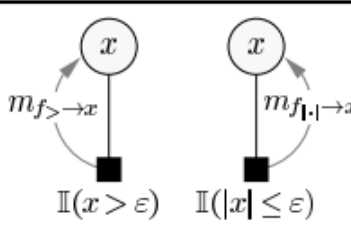
Factor	Update equation
 $\mathcal{N}(x; m, v^2)$	$\pi_x^{\text{new}} \leftarrow \pi_x + \frac{1}{v^2}$ $\tau_x^{\text{new}} \leftarrow \tau_x + \frac{m}{v^2}$
 $\mathcal{N}(x; y, c^2)$	$\pi_{f \to x}^{\text{new}} \leftarrow a(\pi_y - \pi_{f \to y})$ $\tau_{f \to x}^{\text{new}} \leftarrow a(\tau_y - \tau_{f \to y})$ $a := (1 + c^2(\pi_y - \pi_{f \to y}))^{-1}$ $m_{f \to y} \text{ follows from } \mathcal{N}(x; y, c^2) = \mathcal{N}(y; x, c^2).$
 $\mathbb{I}(x = \mathbf{a}^\top \mathbf{y})$	$\pi_{f \to x}^{\text{new}} \leftarrow \left(\sum_{j=1}^n \frac{a_j^2}{\pi_{y_j} - \pi_{f \to y_j}} \right)^{-1}$ $\tau_{f \to x}^{\text{new}} \leftarrow \pi_{f \to x}^{\text{new}} \cdot \left(\sum_{j=1}^n a_j \cdot \frac{\tau_{y_j} - \tau_{f \to y_j}}{\pi_{y_j} - \pi_{f \to y_j}} \right)$
 $\mathbb{I}(x = \mathbf{b}^\top \mathbf{y})$	 $\mathbb{I}(y_n = \mathbf{a}^\top [y_1, \dots, y_{n-1}, x])$ $\mathbf{a} = \frac{1}{b_n} \begin{bmatrix} -b_1 \\ \vdots \\ -b_{n-1} \\ +1 \end{bmatrix}$
 $\mathbb{I}(x > \varepsilon) \quad \mathbb{I}(x \leq \varepsilon)$	$\pi_x^{\text{new}} \leftarrow \frac{c}{1 - W_f(d/\sqrt{c}, \varepsilon\sqrt{c})}$ $\tau_x^{\text{new}} \leftarrow \frac{d + \sqrt{c} \cdot V_f(d/\sqrt{c}, \varepsilon\sqrt{c})}{1 - W_f(d/\sqrt{c}, \varepsilon\sqrt{c})}$ $c := \pi_x - \pi_{f \to x}, \quad d := \tau_x - \tau_{f \to x}$

Figure 3. The update equations for the factor graph [23]

2.1.3 Elo-MMR rating system

The Elo-MMR rating systems [24] is a novel Bayesian rating system which can be applied to multiplayer competitions with distinct ranks as the result. In order to analyze and quantify the skill levels of horses, all ranking records of horses in the past races are aggregated together and stronger horses which win consistently in the past will have a higher skill level. In the experiments shown in the original paper, it gives a more accurate result with a very efficient time complexity than the existing rating systems when the number of players is large enough.

The Elo-MMR rating system is designed with clear goals. The first goal is estimating accurate results in time-efficient manner even though the size of population is large. The second goal is to be incentive compatible. It means that horses' ratings should not have opposite changes to their performance in the races. For example, the horse's rating should not be escalated if it gets a place lower than the place that it got in the last race or vice versa. The third goal is providing a human interpretable rating that the overall skill of a horse can be encapsulated with a single number. One of the reasons for setting the above goals is attempting to avoid the complex mechanism like the message passing in the TrueSkill rating system which takes more time because the message passing process needs to iterate until convergence has no rigorous justification due to the complexity.

Ultimately, the simplicity of the Elo-MMR system enables rigorous analysis of the massive, monotonic, and robust properties as mentioned from its name. The massive property indicates that the computation time is scaled only linearly with increasing size of the population. The monotonic property is equivalent to the incentive

compatible property mentioned in its goal which means stronger horses are always expected to have high ratings. The robust property sets a dynamic bound to the change of the horse's rating so that volatile horses have a larger bound than those consistent horses. As a comparison, Elo-mmr should be better than the Trueskill rating system because Trueskill rating system cannot meet the robustness requirement and intends to achieve the first two properties without rigorous justification.

The races take place sequentially and we denote the series of races as $t = 1, 2, \dots, n$. Then, we denote all horses in the race t as H_t . The i -th horse's skill level at race t is a real random variable denoted as $S_{i,t}$. The performance of the i -th horse in race t is denoted as $P_{i,t}$ and it should have similar value to $S_{i,t}$. We further assume that the difference between performance and skill level for each horse should be independent of its skill level.

The ranking of the race t which is described as the evidence E_t would be responsible for the Bayesian updates. As a result, Elo-MMR calculates the skill level of horse i in race t based on the entire ranking history before race t .

According to the above notations, we can write the joint distribution described by Elo-MMR below,

$$P(S, P, E) = \prod_i P(S_{i,0}) \prod_{i,t} P(S_{i,t} | S_{i,t-1}) \prod_{i,t} P(P_{i,t} | S_{i,t}) \prod_t P(E_t | P_t). \quad (11)$$

The above equation includes one prior distribution and three models.

- $P(S_{i,0})$ represents the initial skill level prior.
- $P(S_{i,t} | S_{i,t-1})$ represents the skill evolution model with previous skill level as information.

- $P(P_{i,t}|S_{i,t})$ represents the performance model with current skill level as information.
- $P(E_t | P_t)$ represents the evidence model with performances of all participating horses as information. It is an indicator function which equals to one if the relative order of performance of all horses in race t P_t is same as E_t . Otherwise, it equals to zero.

2.2 Background knowledge of Transformer

In horse racing, the winner is believed to be a relatively skillful horse which defeats the other relatively weaker horses. Therefore, the dependencies between horses should be captured for doing comparison and prediction instead of treating horses in a single race independently. As our input is a long sequence of information of all horses in a race, we need a model that can handle sequence modelling and dependencies between the information in the input owing to its attention mechanism. Transformer turns out to be a proper network structure fulfilling our requirements and solves our problem more efficiently as compared to convolutional neural network and recurrent neural network.

2.2.1 Transformer

The transformer [25] has an encoder-decoder structure. The encoder in transformer converts input sequences of discrete values to an intermediate sequence of continuous values. Then, the decoder makes use of the intermediate sequence to produce the tokens in the output sequence one by one because the previous token in the output is also the input for producing the next token.

2.2.1.2 Model architecture

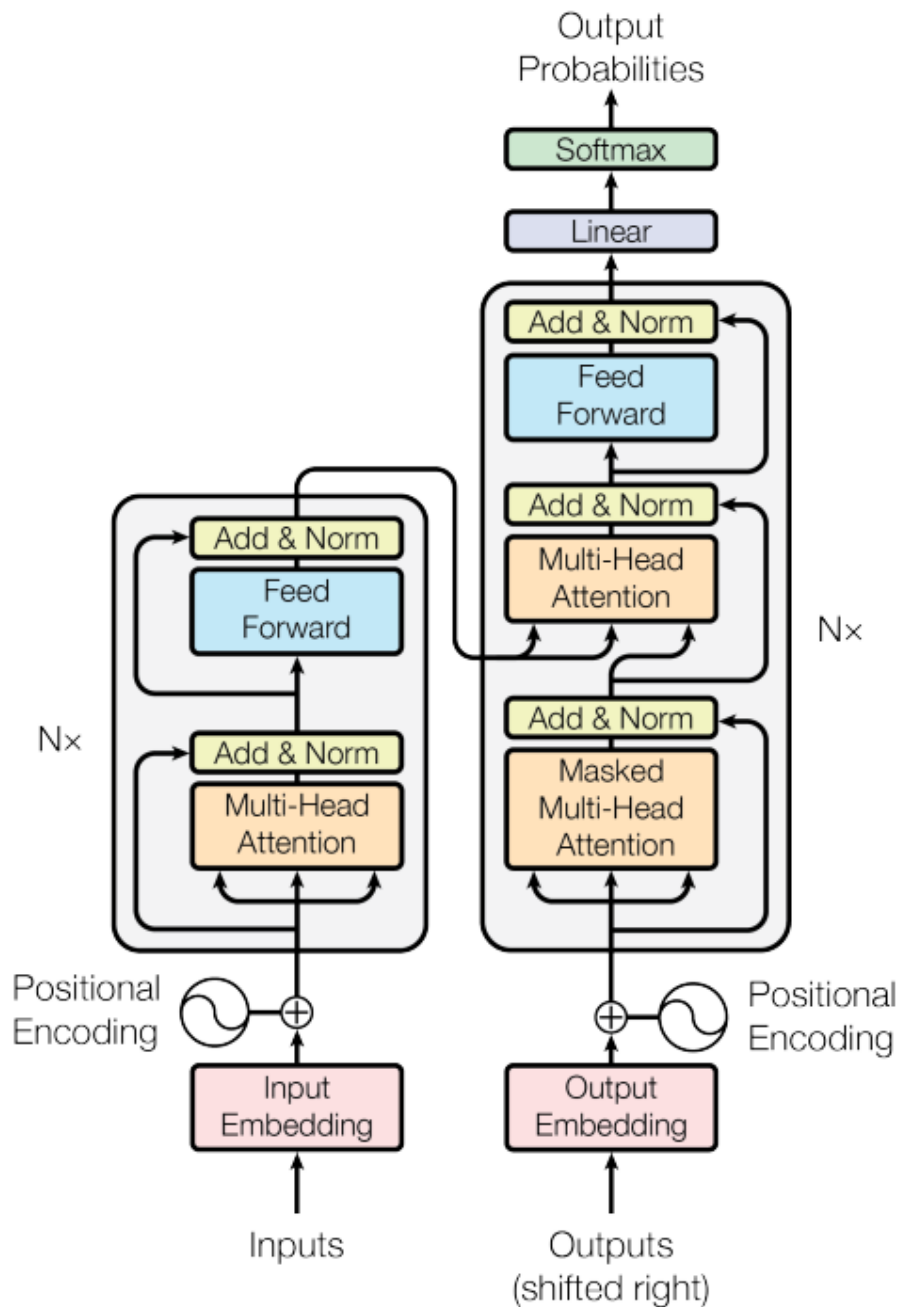


Figure 4. The transformer architecture [25]

In Figure 4, it shows the general structure of a transformer. It contains a stack of self-attention and fully connected layers in core components encoder and decoder. Details is explained in the later sections.

2.2.1.2 Encoder

The encoder is formed by N exactly the same layers while each layer in the stack can be further separated into two sub layers. The input sequence is first embedded through an embedding layer to have dimension d for each token before entering the encoder stack. The input x of the layer enters the first sub layer of the encoder stack which is the multi-head attention mechanism. Then, the original input is added to output of the multi-head attention mechanism, which is fed to the normalization layer, $LayerNorm(x + multi_head_attention(x))$. After that, the output of the normalization layer is passed to a full connected feed-forward layer and residual connection is again employed here so that the normalization layer following the full feed-forward layer is $LayerNorm(x + feed_forward(x))$.

2.2.1.3 Decoder

The decoder is basically same with the encoder except that it has an additional multi-head attention. A mask is introduced to the first multi-head attention in the decoder stack. The purpose of the modification is preventing positions from attending the unread positions and ensuring output at position k can only reference to the output before position k .

2.2.1.4 Attention

In an attention function, the input consists of three vectors which are query, keys and values. Query and keys together undergo a compatibility function to give the weights. Then, the weights are combined with the values to produce the output, a weighted sum of the values.

2.2.1.5 Scaled Dot-Product Attention

Query and keys both have dimension k while the values have dimension v . The weights of values are computed by feeding the division of dot products of the queries and keys by the square root of k to a SoftMax function. Generally, the output is generated with the following formula where Q is the matrix of a set of queries, K is the matrix of a set of keys and V is the matrix of a set of values. The diagram describing the scaled dot-product attention is shown in Figure 5.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{k}}) \cdot V \quad (12)$$

Scaled Dot-Product Attention

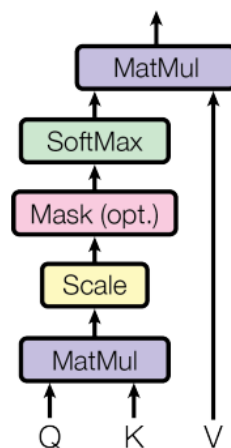


Figure 5. Scaled Dot-Product Attention [25]

2.2.1.6 Multi-Head Attention

We take an approach alternative to input the original queries, values and keys into the single attention function. The queries, keys and values of dimension d are linearly projected to h different versions of queries, keys and values with dimensions k , k and v respectively. These different versions of queries are parallelly processed with the Scaled Dot-Product Attention and each of them will produce the values vectors of dimension v . Finally, we concatenate the values outputted from the Multi-head attention and they are projected as the final values. The following functions describe the process in mathematical way. The diagram describing multi-head attention is shown in Figure 6.

$$head_i = Attention(QW_i^Q, KW_i^K, VW_i^V) \text{ for } i = 1, \dots, h \quad (13)$$

$$MultiHead(Q, K, V) = Concat(head_1, \dots, head_h) \cdot W^O \quad (14)$$

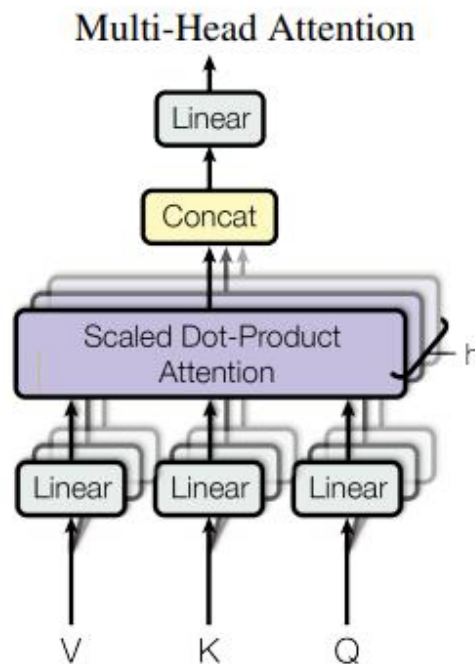


Figure 6. Multi-head Attention [25]

The attention mechanism helps us in capturing the dependencies between horses because the self-attention layers in the encoder allows each position in the encoder to attend to every position in the former layer of the encoder.

2.2.1.7 Positional Encoding

The information regarding the relative and absolute position of each information in the sequence is inserted because the Transformer does not have recurrence and convolution. Therefore, positional encodings of dimension d are added to the embeddings of the input before it enters the stacks for preserving the ordering and position information. It uses two different functions for encoding the odd and even dimension position.

Let i be the dimension and pos be the position,

$$Postional_Encoding(pos, 2i + 1) = \cos(pos/10000^{2i/d}) \quad (15)$$

$$Postional_Encoding(pos, 2i) = \sin(pos/10000^{2i/d}) \quad (16)$$

2.3 Evaluation Strategy

We want to evaluate the model in the profit-making aspect and accuracy aspect after experiments has been done on the horse racing datasets. We propose the following strategies to decide the performance and effectiveness of adapting the transformer model with rating of horses as a replacement of winning odds in the input.

2.3.1 Random betting (Profit-making aspect)

In random betting, we randomly select a horse number from all the participating horses. If the selected horse wins, we get back our bet multiplied by the win odd of the winning horse. Otherwise, we lose our bet. It is assumed to be the worst betting strategy because no knowledge is learnt from the data before doing the prediction.

2.3.2 Lowest odd betting (Profit-making aspect)

In Lowest odd betting, we always guess the horse with the lowest win odds as the winner. If the prediction is correct, we gain the amount of bet times the win odd of the winning horse. Otherwise, we lose our bet. It is believed that the lowest odd betting is much better than random betting because the win odds of horses changes according to public opinion due to pari-mutuel betting and thus it reflects the public intelligence. As the public use their knowledge and experience from the former races in doing the prediction, we assume this strategy surpasses the random betting.

2.3.3 Multilayer perceptron prediction (accuracy aspect)

The multilayer perceptron consists of multiple fully connected feed forward layers is a simple structure of the neural network for doing the prediction. No specific assumption is made about the properties of the input and we think it should have a

lower accuracy than our model.

2.3.4 Transformer without rating in the input (accuracy aspect)

From the study of previous FYP students, the winning odds of horses are the important features in doing the prediction [9]. Since we use the ratings of horses to replace the win odds, we want to show that the ratings are equivalent to the winning odds that they could boost the accuracy of the prediction.

Chapter 3

Data Preparation

3.1 Data Collection

Although past horse racing records could be bought directly via websites hosted by companies such as <https://horseracedatabase.com/> and <https://www.hkhorsedb.com/> which has database storing the historical data, we prefer to collect the data by ourselves because of the expensive prices.

In addition to the financial consideration, writing web crawlers to collect data by ourselves provides us more flexibility to the choice of data because we are freely to retrieve the data that we want by simply configuring our own web crawler. In this project, a web crawler was written for collecting data on the HKJC official websites within a given period. The user can specify the start date and end date so that the crawler will automatically collect the horse race record and horse information from the start date to the end date automatically.

3.2 Data Description

There are total 9191 race records in our dataset dated from June 6, 2008 to October, 17 2021. Every row is a race record storing the attributes of a race such as the venue, class, and distance. All races were hosted by the HKJC and taken place in Hong Kong. The information about the horses appeared in the race records were also collected from the HKJC official database.

3.2.1 Racing Record

Table 3 below shows the features of our race record and their detailed information.

Feature	Description	Types	Values
Date	Date of the race	Index	-
Race_id	The id of the race	Index	-
Venue	Location of the race	Categorical	-
Season_race_no	The number of race in the season	Categorical	In range [1 , 800]
Horse_class	Class of the horses Stronger horses compete in high race class	Categorical	1 - 5
Distance	The distance of the race	Categorical	1000, 1200, 1400, 1600, 1650, 1800, 2000, 2200, 2400
Going	Condition of the lane	Categorical	>= 10 distinct values
Course_track	The lane of the race	Categorical	A, A+3, B, B+2, C, C+3
Course_track_code	Description about the lane	Categorical	TURF, ALL WEATHER
Horse_i_place	The rank of horse i in a race	Categorical	14 distinct values
Horse_i_number	The number of horse i in a race	Categorical	14 distinct values
Horse_i_name	The name of horse i	Categorical	> 5000 distinct values
Horse_i_jockey	The name of jockey	Categorical	> 200 distinct value
Horse_i_trainer	The name of trainer	Categorical	> 200 distinct value
Horse_i_actual_weight	The total weight of horse i and gears	Float	-

Horse_i_declared_weight	The weight of horse i	Float	-
Horse_i_finish_time	The time when horse i finishes the race	Float	-
Horse_i_win_odds	The win odd of horse i	Float	-

Table 3. Feature description of race records

3.2.2 Horse Information

Since the horse's information was useful indicator of the horse's performance in a race, we gathered 6642 horses which all participated in the races recorded in our dataset for doing comparison between horses in a particular race. Table 4 shows the traits of a horse in our horse dataset.

Feature	Description	Types	Values
Horse_origin	The place of birth	Categorical	>10 distinct values
Horse_age	The age of horse	Categorical	In range [3, 10]
Horse_color	The color of skin	Categorical	>6 distinct values
Horse_sex	The gender of horse	Categorical	Colt, Gelding, Mare etc.
Horse_1st_place_frequency	The frequency of getting 1 st place	Categorical	In range [0,20]
Horse_2nd_place_frequency	The frequency of getting 2 nd place	Categorical	In range [0,30]
Horse_3rd_place_frequency	The frequency of getting 3 rd place	Categorical	In range [0,30]
Horse_total_race	The total count of horse's participation	Categorical	In range [0,100]
Horse_sire	Name of horse's father	Categorical	-
horse_dam	Name of horse's mother	Categorical	-
horse_dam's_sire	Name of horse's maternal grandfather	Categorical	-

Table 4. Feature description of horse records

3.3 Data Analysis

Among all the features describing races and horses, it is believed that not all features are equivalently important in doing the prediction of horse's performance. Therefore, we would like to study the influences of features on the result of races. In the data analysis, we first investigate the distribution of the selected categorical feature given the winning horses and then look at the likelihood $P(X = x | Y = y)$ where x is the selected categorical feature and y is the winning horse. Finally, we examine the performance of horses by the correlation between numerical features especially the finish time and win odds as a horse usually performs well if it finishes the race in a shorter time and has a low win odd.

3.3.1 Categorical Features

3.3.1.1 Age

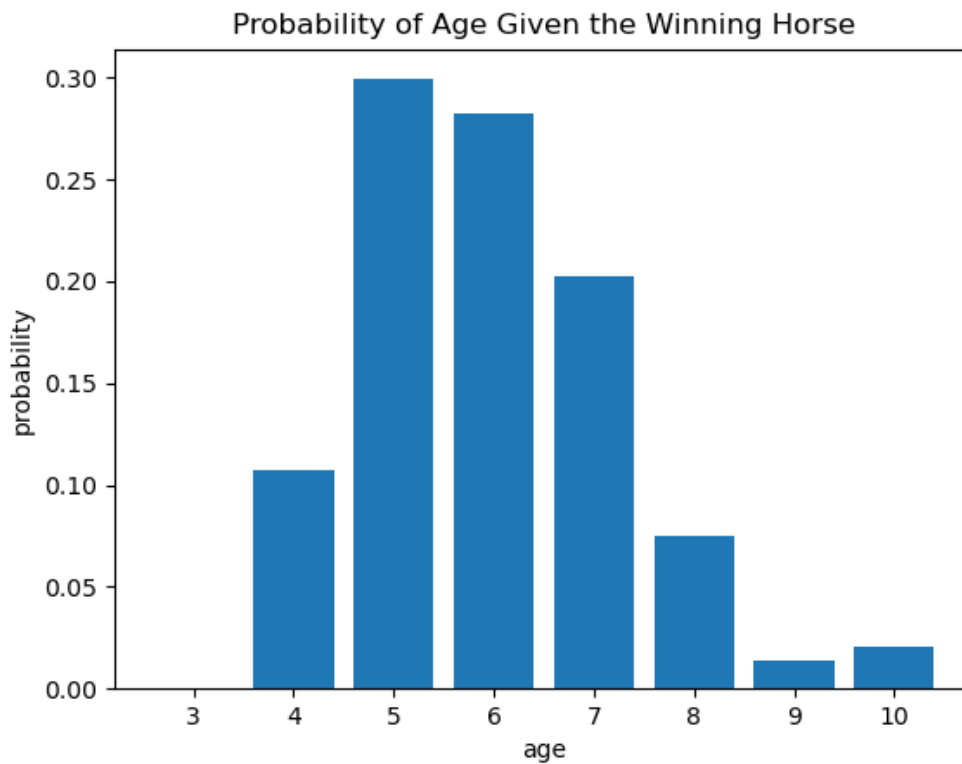


Figure 7. The distribution of age given the winning horse

Our data demonstrates the declining performance of horse with increase age as shown in Figure 7. Among all the winning horses, more than 50% are horses aged between 5 and 6 as horses' optimal body weight and skeleton are reached at 4 or 5 years old [26]. The number of winning horses decreases substantially after the age of 5. It implies that the overall performance of majority horses reaches their peak when they are 5 or 6 years old and then decline due to the decrease in stamina, speed and power brought by aging. The horses aged between 3 and 4 accounts for approximately 11% in the winning horses. One explanation for fewer winning horses with lower age is that they have not joined enough competitions to be very skillful and they are still growing.

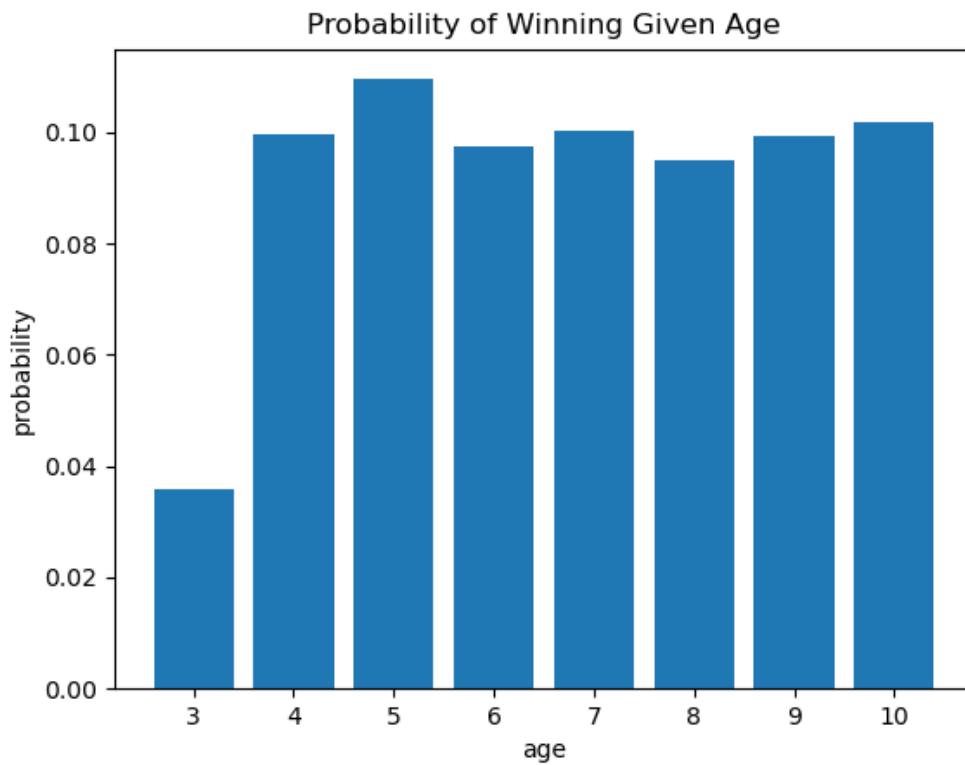


Figure 8. The distribution of winning horse given age

Although the likelihood exhibits the tendency to choose horses aged between 5 and 6 to be winners, we observe that horses have similar probability of winning at around 10% for all ages except 3 which is shown in Figure 8. It suggests that the winning condition cannot be determined solely on the age of individual horse.

3.3.1.2 Origin

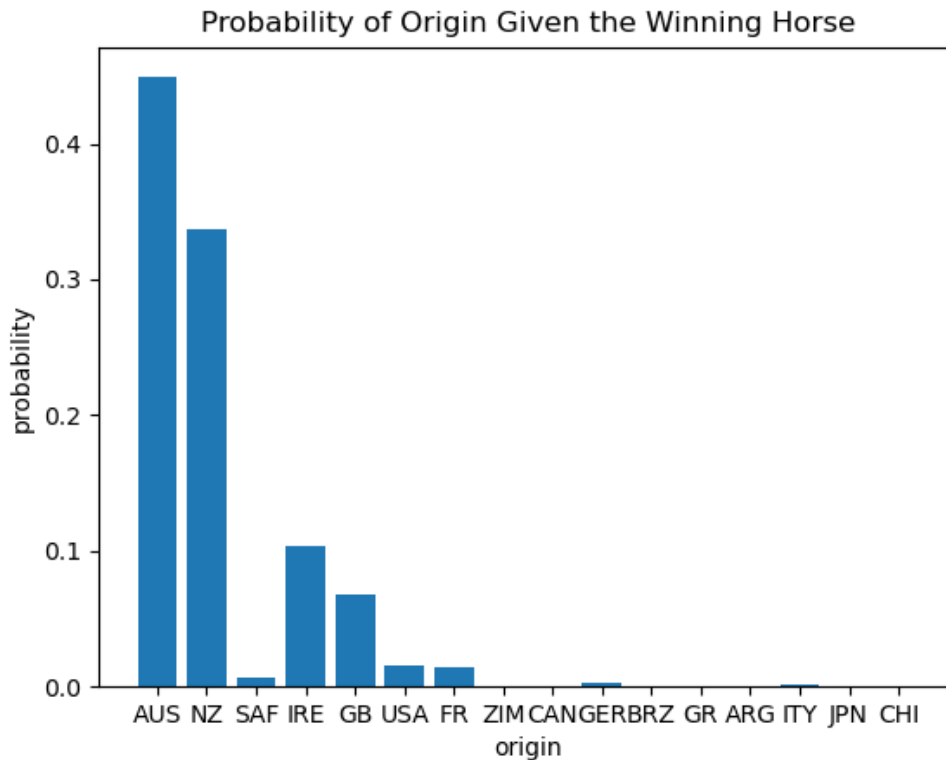


Figure 9. The distribution of origin given the winning horse

Most winning horses were born in Australia or New Zealand as shown in Figure 9. It reflects that horses which were born in Australia or New Zealand are usually perform better than horses coming from the other countries. This information is useful when we want to do a simple classification to identify all horses with various origins in a single race into two classes which are likely to win and unlikely to win. In this situation, horses from Australia or New Zealand will be classified as likely to win while horses from other countries will be classified as unlikely to win.

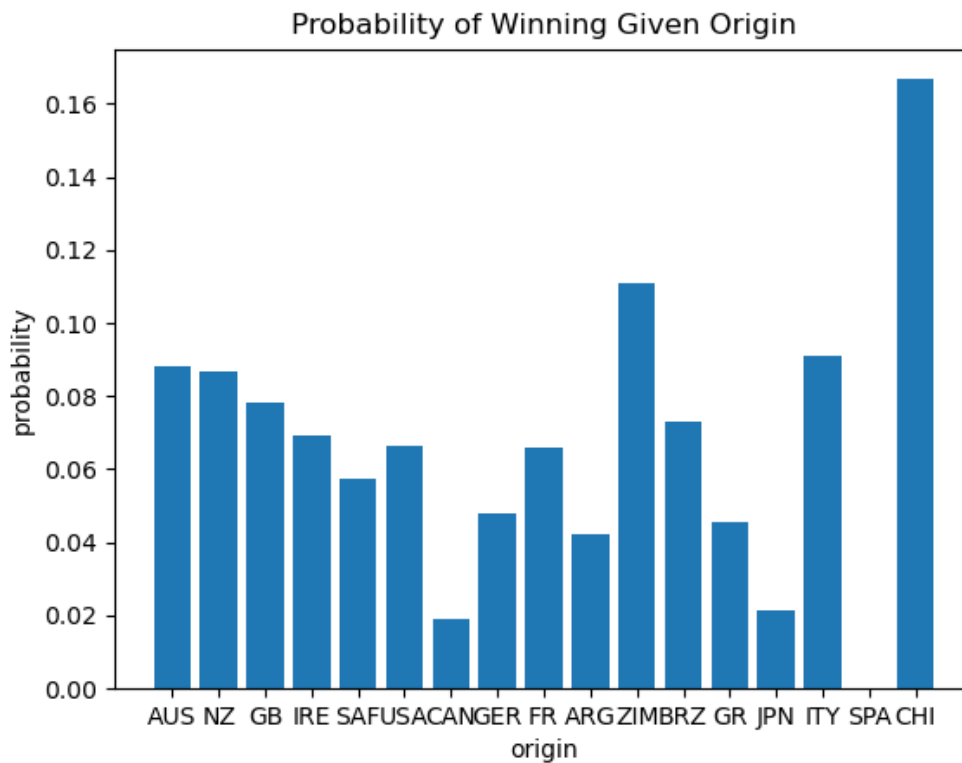


Figure 10. The distribution of winning horse given origin

Since most horses in horse race are imported from Australia and New Zealand, this may create bias to the winning distribution that horses from Australia and New Zealand are usually winners. After conditioning the winning probability by the origin, we see that horses Republic of Zimbabwe and República de Chile. Nevertheless, the number of horses coming from Republic of Zimbabwe and República de Chile is very small while the number of horses from Australia and New Zealand is huge. Figure 10 shows that horses Australia and New Zealand are still likely to be the winner in real case when compared to other countries except Republic of Zimbabwe and República de Chile.

3.3.1.3 Color

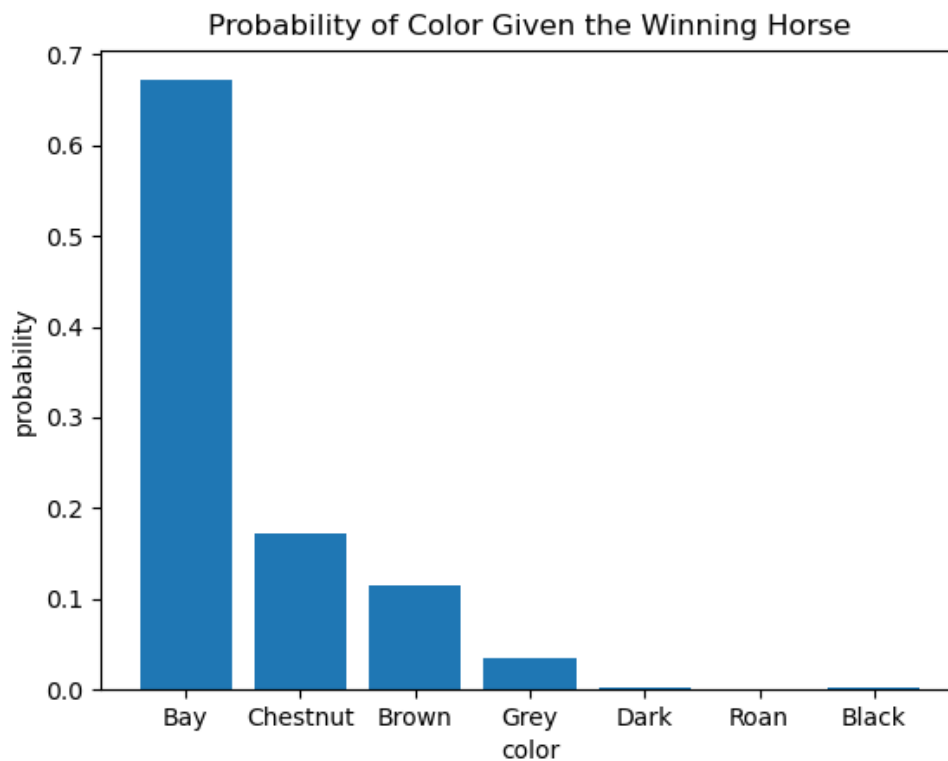


Figure 11. The distribution of color given the winning horse

More than 65% winning horses have skin color Bay as shown in Figure 11. The second most color is Chestnut with 17%. The remaining colors like Brown, Grey, Dark, Roan and Black only constitute a small portion in the winning horse. The large distribution of color Bay in winning horse suggests that color would be a good choice for being the early decision boundary in machine learning method that adopt the greedy approach such as decision tree.

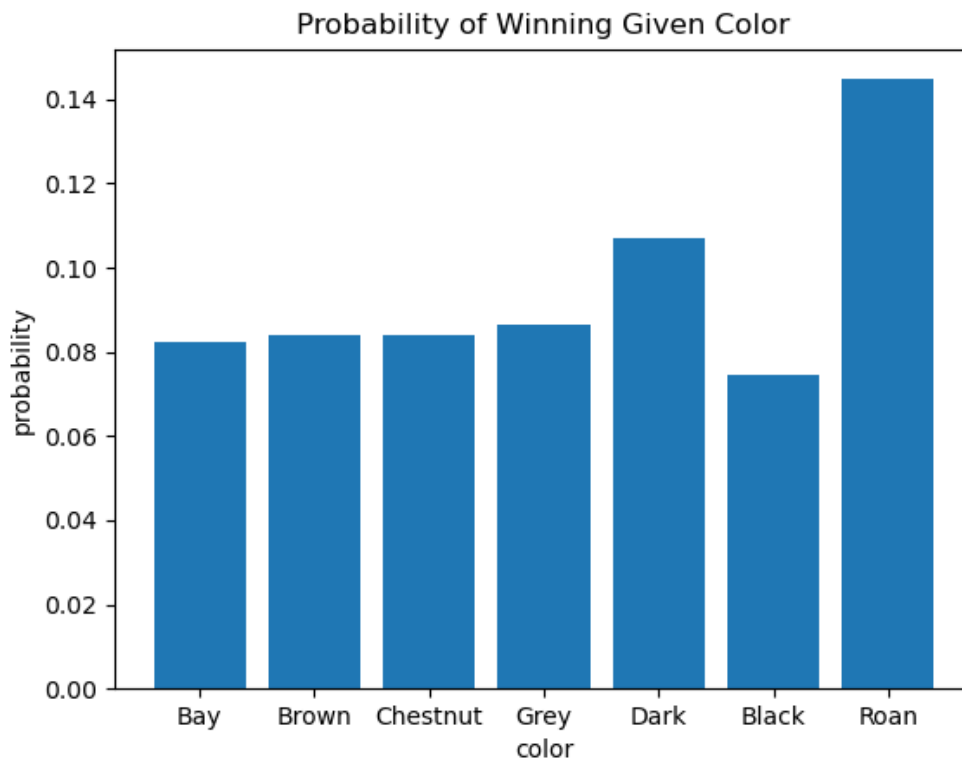


Figure 12. The distribution of winning horse given color

When the winning probability is conditioned on the color, the advantage of horses with color Bay loses while those colors which are less likely to appear in winning such as Dark and Roan horses surpass. Also, the winning probability of horse with color Bay is the second lowest in Figure 12 and it implies that our observation from Figure 11 is biased as a large portion of horses in horse race have skin color Bay.

3.3.1.4 Sex

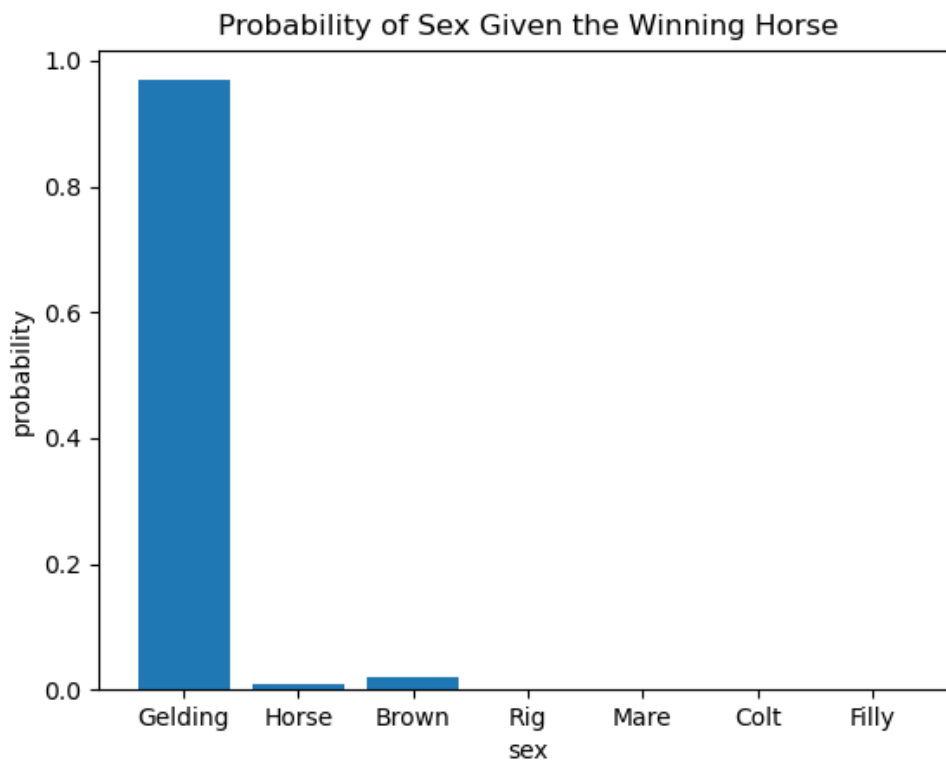


Figure 13. The distribution of sex given the winning horse

The likelihood distribution of sex in winning horses is dominated by the sex Gelding. Over 97% of winning horses with sex Gelding as shown in Figure 13. The sex Horse and Brown only constitute a very small portion in the winning horse with approximately 3% in total. This likelihood is highly biased because almost all horses in horse race has are with Gelding and therefore this feature should have extremely few impacts on the race result.

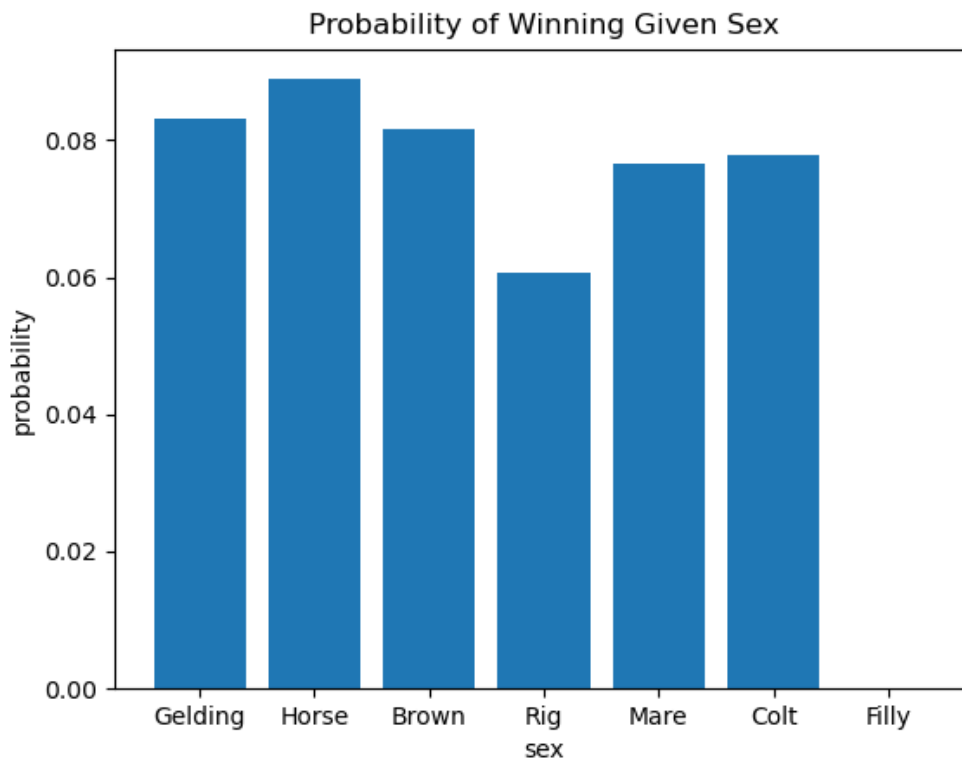


Figure 14. The distribution of winning horse given sex

The conditional probability of winning given the sex of the horses confirms our assumption that the horse with sex Gelding is the most likely the winner is flawed because the probability of winning given the sex is Gelding has similar value with the probability of winning given other sex. From Figure 14, we are more confident that the horse with sex Horse will win the race as it has the highest conditional probability of winning among all horses with other sex.

3.3.1.5 Draw

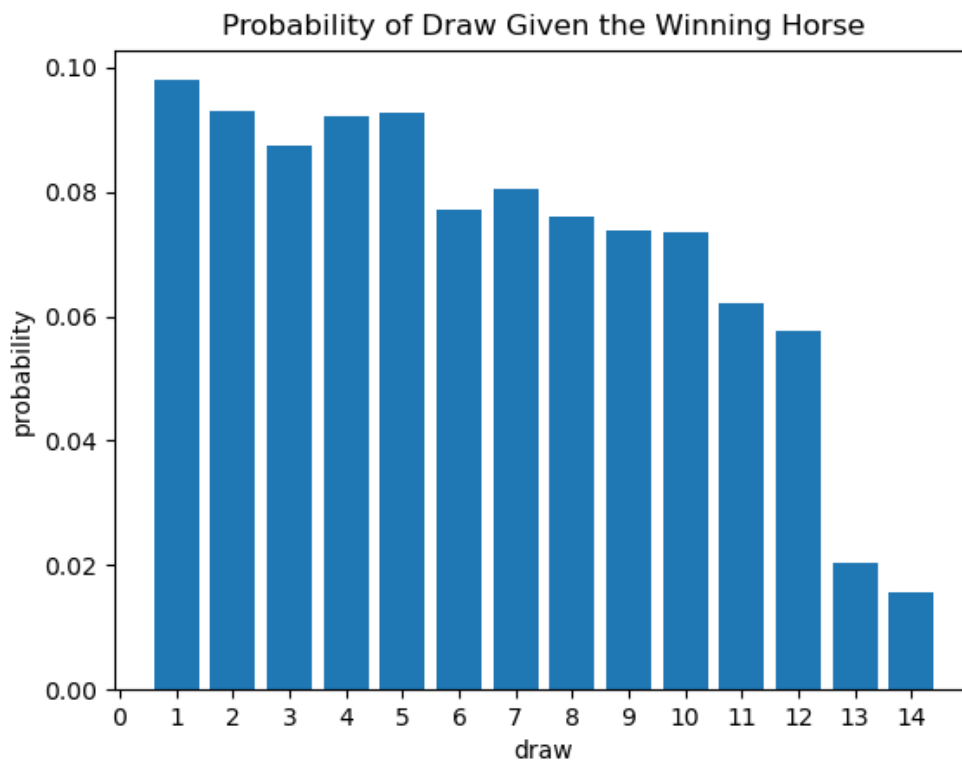


Figure 15. The distribution of draw given the winning horse

Horses with smaller draw number are considered to be opportune in horse racing because they are arranged towards the center of the circular track as shown in Figure 15. The running distance of those horses is thus relatively shorter than horses with larger draw number which means horses with smaller draw number need shorter time in finishing the race. Our data agrees with our assumption about the advantage of smaller draw number since there is a declining proportion of winning horse with increasing draw number.

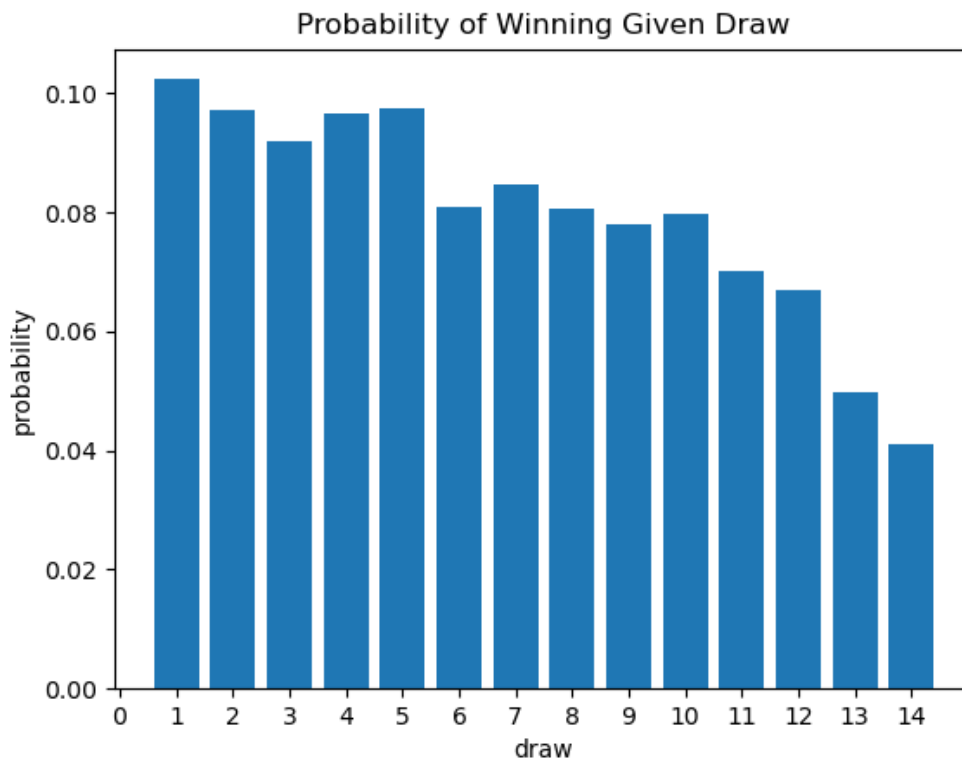


Figure 16. The distribution of winning horse given the draw

The conditional probability of winning given the draw has similar shape with the probability of draw given the winning horses as shown in Figure 16. Hence, the fact that the horses with smaller draw number are more likely to be the winner is assured. However, the horses with large draw number also win in some races so we the other factors should be considered in determining the winning probability of them.

3.3.2 Numerical Features

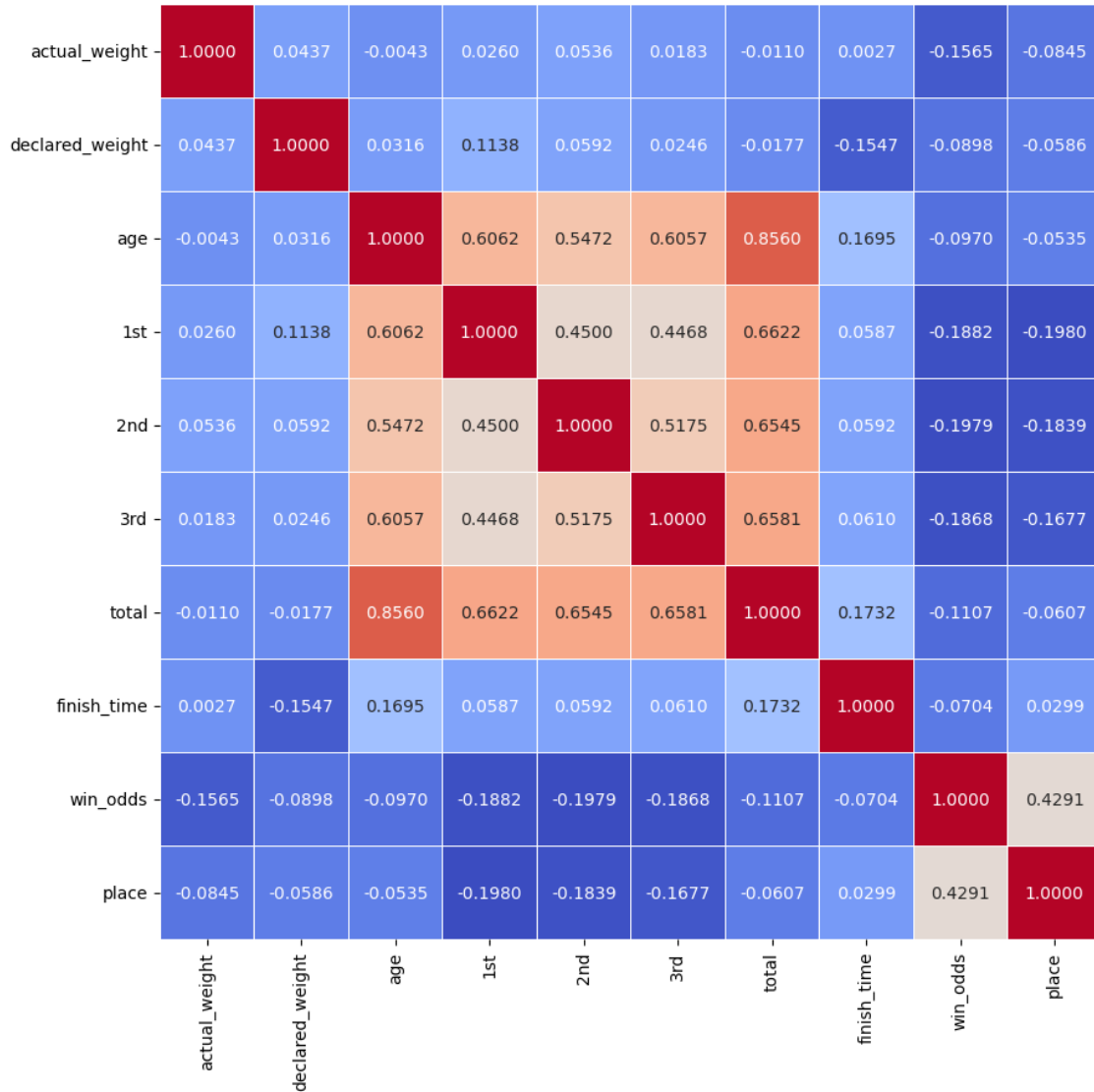


Figure 17. The correlation matrix of numerical features

In analyzing the numerical features, we investigate the correlation between each pair of horse features in our dataset from Figure 17. The cell of darker color in the correlation matrix implies a stronger correlation or vice versa. Some essential horse features that have strong influences on the result of a race are selected for the following discussion.

3.3.2.1 Frequency of 1st Place

We examine the row of feature frequency of 1st place which is the count of being a winner in a race. It has a significant correlation with the frequency of 2nd place and 3rd place which are 0.4500 and 0.4468 respectively. As the correlation coefficients are positive, it infers the positive relationship between frequency of 1st place, 2nd place and 3rd place. The relationship matches out expectation that a horse with good performance in the past, getting first three places in the past race, often performs well in the next race. Notice the negative correlation of -0.1882 between the frequency of 1st place and the win odds. The public has the same opinion about the consistent performance of horses in future races, so they tend to bet the horse with large count of 1st place, and it results in a lower win odd of the horse owing to the pari-mutuel betting system. Besides, the consistent performance is proven by the negative correlation of -0.1980 between the count of 1st place and the places that the horse gets in races.

3.3.2.2 Finish time

The finish time is a measure of horse performance since we assume stronger horse will finish a race in shorter time. This motivates us to examine the correlation of the finish time and other horse features. A negative correlation of -0.1547 between finish time and declared weight is shown in the correlation matrix. The handicapping policy by the HKJC adds weights to well performed horses and the declared weight is increased so that the chances for horses of worse performance are increased [11]. From our data, we see that the policy is not effective enough because the well performed horses with more declared weights still have a shorter finish time. On the

other hand, there is a positive correlation of 0.1695 between the finish time and the age. This agrees with our analysis in the age of horse that the performance of horses declines with age.

3.3.2.3 Win odds

The win odds of horses reveal the general guess of the public since the win odds of horses change with the amount of bet. The more the popular the horse, the lower the win odd of the horse. From the correlation matrix, win odds has negative correlations of -0.1882, -0.1979 and -0.1868 between the frequency of 1st place, 2nd place and 3rd place respectively. This implies that the frequency of 1st place, 2nd place and 3rd place guide the public to make the decision. The larger the number about this statistic, the lower the win odds. Another discovery is the positive correlation of 0.4291 between the win odds and the place. This shows that the public intelligence is accurate in some sense. For example, if the public do not think the horse will win, they will not bet on it and the horse will have a very high win odd. If the public intelligence is accurate enough, the horse with high win odd should not perform well and get the small place number.

3.4 Data Preprocess

The raw data scraped from websites are not clean and well organized, so they should be preprocessed into a desirable format before feeding them into our neural network models. We had done the following four steps data imputation, data encoding, input normalization and rating generation on our dataset before starting the experiments.

3.4.1 Data Imputation

In the data collection process, we inevitably encounter the network error such as link rot or unresponsive server especially when the target data is old. This happened when we collect the data of some retired horses which took part in the races before year 2010. Hence, a small part of horse data about those retired horses is missing in our data set. However, it is unadvisable to omit or remove the horse records with missing information as the records may affect the quality in the knowledge extraction procedure and biased estimation would be made when doing the analysis [27].

Addressing the missing information, we decide to do data imputation on our dataset by using the k nearest neighbors method. First, we extract all complete horse records without missing values. Then, we place the missing value in an incomplete record by looking for its k nearest neighbors in the complete horse records. The value filled in the missing part will be the mean of neighbors if the type of feature is numerical. Otherwise, we do a majority vote on the neighbors and place the most common categorical value in the missing part [27].

Instead of implementing the k nearest neighbors, we invoked the KNN Imputer from Scikit Learn library to ensure the simplicity and correctness.

3.4.2 Data Encoding

The input of our neural network models must be numerical but some of our data are categorical. For instance, the horse's name, jockey name, distance and course track are categorical values. For this reason, we need to transform our categorical data into numerical by data encoding.

One simple method is converting the categorical data in form of one hot encoding in which we use k binary features to represent a categorical feature of 2^k classes. The value of the binary feature is either 1 or 0. However, the dimension of our input will be increased drastically for representing all categorical data and it requires extra memory and more computational time for the training [28].

Ordinal Encoding scheme is applied to our data as we do not want additional memory usage and extra computational time due to the one hot encoding. In this scheme, a unique integer means a category and no new columns are added so the dimension of the data is the same as the original. Furthermore, the order of ordinal variables is preserved in this scheme [29]. For example, the feature place which has 14 classes representing the ranks of horses in a race. We encode the horses of higher rank with a smaller integer to preserve the ranking order.

In the implementation part, we invoked the Ordinal Encoder from the Scikit Learn Library to maintain the simplicity of our code.

3.4.3 Normalization

Normalization of the input was done before the training of our models. It was shown that normalization of input data can produce a better result and speed up the training process. On one hand, values of all variables are scaled to have the same range which saves the effort for backward propagation in changing the weight of variables. On the other hand, the same scale of all variables balances the focus of error minimization in the weight correction algorithm so that importance of variables is distributed evenly to avoid bias [30].

We use z-score normalization which takes the mean and standard deviation of each feature in column direction of our input vector and use that information to compute the values for the corresponding feature. The formula is shown as below,

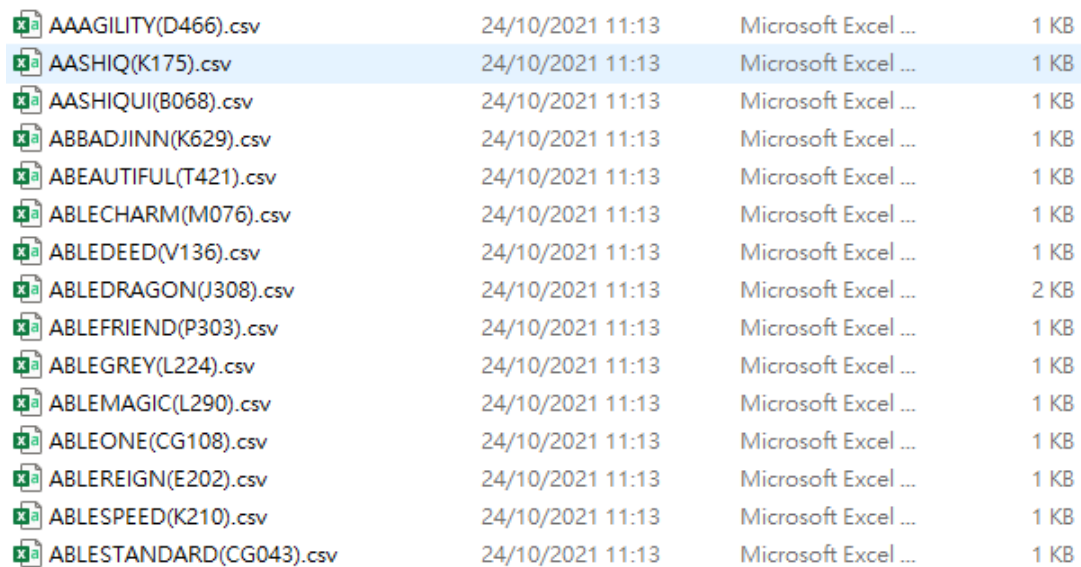
$$x'_i = \frac{x_i - \bar{x}_i}{\sigma_i}. \quad (17)$$

where x'_i is the computed value, \bar{x}_i is the mean of the feature and σ_i is the standard deviation of the feature.

3.4.4 Rating Generation

Rating about the horse performance is one of the focuses in this project. Nonetheless, the ratings mentioned in the methodology do not exist on the HKJC websites and we need to calculate those ratings with the information provided by our dataset.

In rating generation, we mapped horse records to race records under the guidance of the horse names in race records so that we obtain the race records with horse records ordered from 1st place to the last place. Then, we reformatted this each record into a json file and named each file with a number. The smaller the number, the older the race. After that, we invoked a rating computation library [31] and used all json files as the input. Then we had a list of horses with their rating in each race which was then merged to our original dataset. The list of json file is shown in Figure 18. The content of rating file for each horse is shown in Figure 19.



AAAGILITY(D466).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
AASHIQ(K175).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
AASHIQUI(B068).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABBADJINN(K629).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABEAUTIFUL(T421).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLECHARM(M076).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLEDEED(V136).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLEDRAGON(J308).csv	24/10/2021 11:13	Microsoft Excel ...	2 KB
ABLEFRIEND(P303).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLEGREY(L224).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLEMAGIC(L290).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLEONE(CG108).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLEREIGN(E202).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLESPEED(K210).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB
ABLESTANDARD(CG043).csv	24/10/2021 11:13	Microsoft Excel ...	1 KB

Figure 18. The list of json files for rating computation

	A	B	C	D	E
1	contest_index	rating_mu	rating_sig	perf_score	place
2	8315	1329	174	1294	13
3	8372	1373	133	1421	12
4	8444	1337	114	1270	14
5	8515	1336	102	1332	12
6	8586	1305	95	1208	14
7	8704	1283	90	1211	11

Figure 19. The rating of horse

Chapter 4

Methodology

4.1 Methodology Overview

The winning odds captures the relative expected performance of horses in a race because bettors tend to bet on a relatively stronger horse and the pari-mutuel betting setting will therefore decreases the winning odd of the horse with better expected performance. Moreover, previous final year project students in LYU1805 illustrated the significance of winning odds in doing horse race prediction [10]. Finish time is also an important metric to evaluate the relative performance of horses since stronger horses can finish the race in a shorter time. However, both winning odds and finish time should be excluded from the feature list because our ultimate goal is estimating the winning odds of horses and we do not know the finish time of horses when we do prediction regarding new races. Therefore, we decide to find another metric to help us figure out the relative performance of horses.

Rating systems estimate the relative skill level of horses based on their historical performance. As we can easily assess a horse's past racing record from the HKJC website, we apply rating systems here to calculate the relative skill point of horses and we wish the ratings can replace the effect of winning odds in our prediction.

Rating systems have different underlying assumptions in calculating the relative skill point and we want to see which rating system best represents the relative skill point of horses, so we are going to experiment on three different rating systems and find the one which produces the best result.

Besides the relative skill level of horses, winning odds also rely on the dependencies between horses' attributes. The attributes of horses in a race are not independent in our context and thus the probability of winning for each horse should be conditioned on attributes of all participating horses. Simple models such as linear regression and a single decision tree are not suitable for this problem because the relationships between the attributes of horses are sophisticated and cannot be easily captured by these simple models. So, models that can learn complex non-linear relationships and dedicated for referencing all attributes of horses in estimation should be selected for the prediction. For this consideration, multilayer perceptron and transformer are chosen to be the models. The diagram representing the methodology is shown in Figure 20.

Therefore, we are going to experiment on the combinations of rating systems and the selected neural network architectures to see whether they can compensate the exclusion of winning odds in prediction. Then, we will compare the results of different combinations and evaluate the performance of them by using the evaluation strategies proposed in section 2.3.

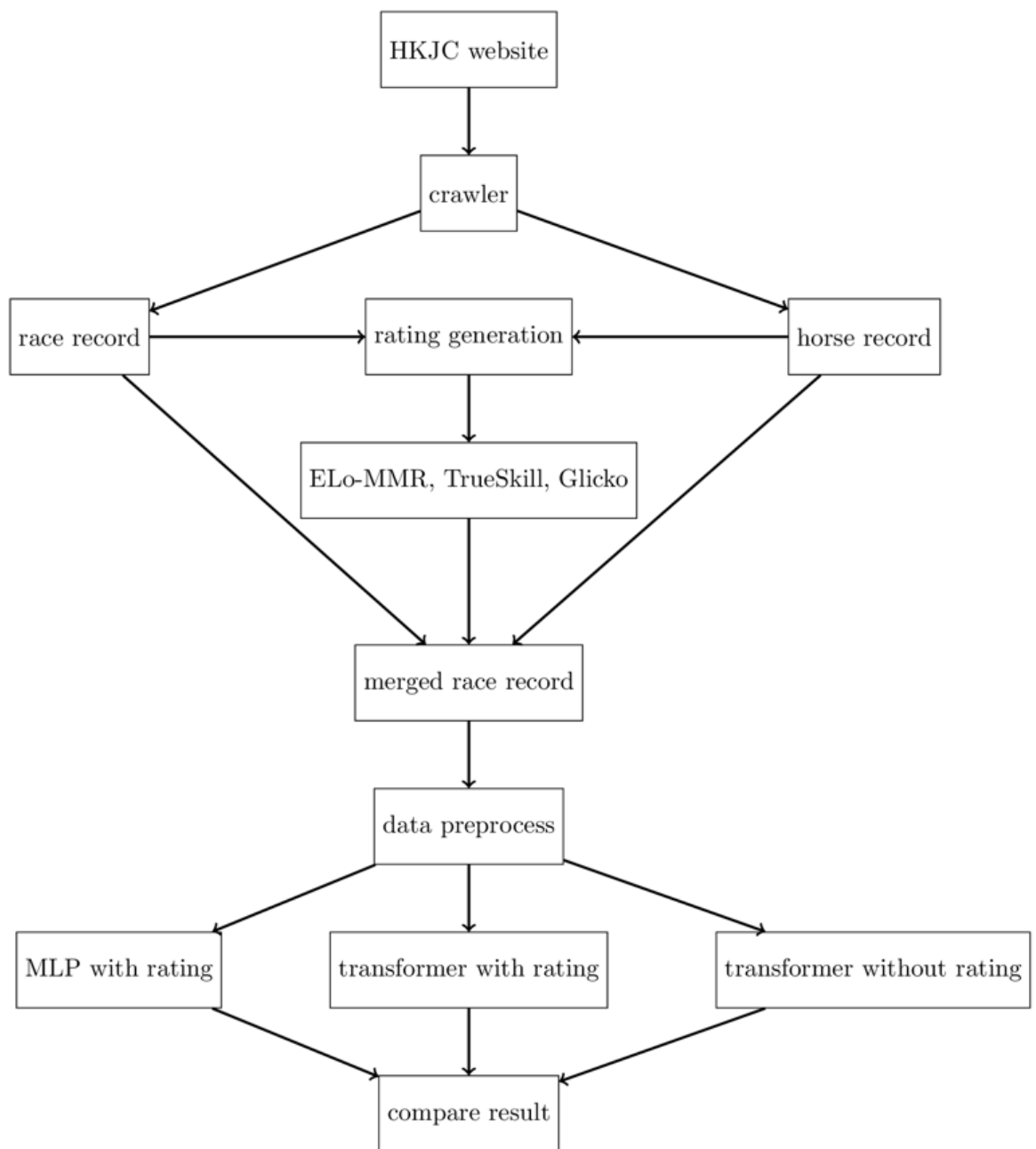


Figure 20. The diagram describing the methodology

4.2 Model Design

4.2.1.1 Multilayer Perceptron Classification

The number of classes in our multilayer perceptron equals to the number of horses in the race. For instance, there will be 14 classes if the race has 14 participating horses, and each class corresponds to a horse number. The input is a race record joined with horse records according to the horse names listed in the race record. The output of the neural network is a vector consisting the values resembling the probabilities of winning of horses and classification is done based on these values. If the horse with horse number 7 wins the race and our multilayer perceptron predicts it correctly by giving it the highest value in the vector, the model assigns this input to class 7.

4.2.1.2 Multilayer Perceptron Architecture

The multilayer perceptron has total 5 linear layers. The first linear layer is the input layer which takes the values of the input vector. There are 3 hidden linear layers with number of neurons at range 100 – 400 for increasing the sensitivity of model in the learning process [32]. The output of each hidden linear layers has to be passed through the ReLu activation function which determines the activity of the neurons. For the second the and the third hidden layers, dropout layers are inserted for the purpose of regularization which helps the model avoid overfitting by randomly losing connections between neurons in the training process [33]. The last linear layer is the output layer storing the outcome. We pick cross entropy function and stochastic gradient descent to be the model's loss function and optimizer. The diagram of the model is shown in Figure 21.

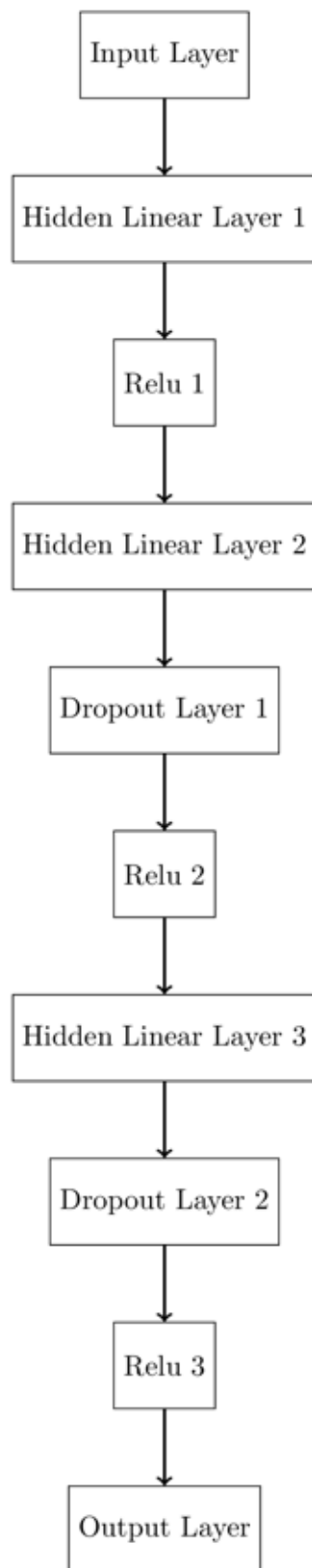


Figure 21. The Multilayer perceptron architecture

4.2.2.1 Transformer Classification

As our ultimate goal of designing two neural network models is comparing the effect of using different neural network architectures in horse racing prediction, we decide to have similar setting regarding the input and output in transformer model as the multilayer perceptron. Therefore, the input of the transformer classification is joined race record and horse records, and the output is the horse number belonging to the winning horse.

4.2.2.2 Transformer Architecture

The transformer model does not use decoder [25] mentioned in the original paper because the output of our classification problem is a single number instead of a sequence. We partition our model into three stages. The first stage is about data formatting of the input vector. We use an embedding layer to increase the dimension of each feature which mimics the word embedding in natural language processing. Then, we use a position embedding layer to remember the position of each feature as position is meaningful in our input data which features of one horse are in closer distance than other features. Next, the processed input enters the encoder of a transformer for learning the dependencies between features. The output of the encoder is sent to a simple fully connected feedforward network consisting 2 hidden linear layers and a output layer. We pick cross entropy function and stochastic gradient descent to be the model's loss function and optimizer. The diagram of the model is shown in Figure 22.

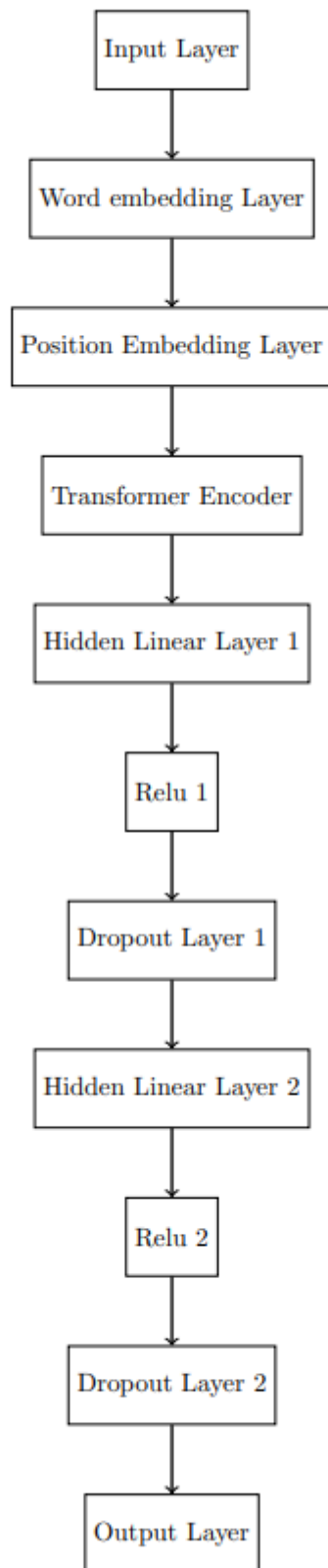


Figure 22. The Transformer architecture

Chapter 5

Experiment and Result

In our experiment, we have trained three models with changes in input features and the neural network architecture. The first model is the multilayer perceptron classification model with inputs including ratings. The second model is the transformer classification model with inputs excluding ratings. The third model is the transformer classification model with inputs including rating. We want to study whether the third model achieves better result and therefore we use the results of the first and the second models to be the reference when evaluating the performance of the third model.

5.1 Input Data

We separate the most recent 688 horse races between 9 December 2020 and 10 October in 2021 from our original horse race dataset for testing. The remaining 8503 horse races are used in the training process. Splitting the training data and testing data randomly is inappropriate in our context because we are more interested in correct predictions of new races and the past races having retired horses should not be involved in the test data when we want to evaluate the performance of our models in predicting the new races.

We formulate each race as a single input after data preprocess as shown in Table 5. All information about a race including the conditions of the track, attributes of horses and ratings are packed into a row in our input matrix. This ensures that the neural network receives sufficient data when predicting the winner horse in a race. For further

comparison about different combinations of neural network architectures and data, a few columns in the input matrix are discarded to study the effect of the discarded features.

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 number of horse i in a race
Horse_i_name	The name of horse i
Horse_i_jockey	The name of jockey
Horse_i_trainer	The name of trainer
Horse_i_declared_weight	The weight of horse i
Horse_origin	The place of birth
Horse_age	The age of horse
Horse_color	The color of skin
Horse_sex	The gender of horse
Horse_1st_place_frequency	The frequency of getting 1 st place
Horse_i_total_race	The total count of horse's participation
Horse_i_rating	The rating of the horse

Table 5. The schema of the input data

5.2 Results

5.2.1 multilayer perceptron classification model

5.2.1.1 Accuracy

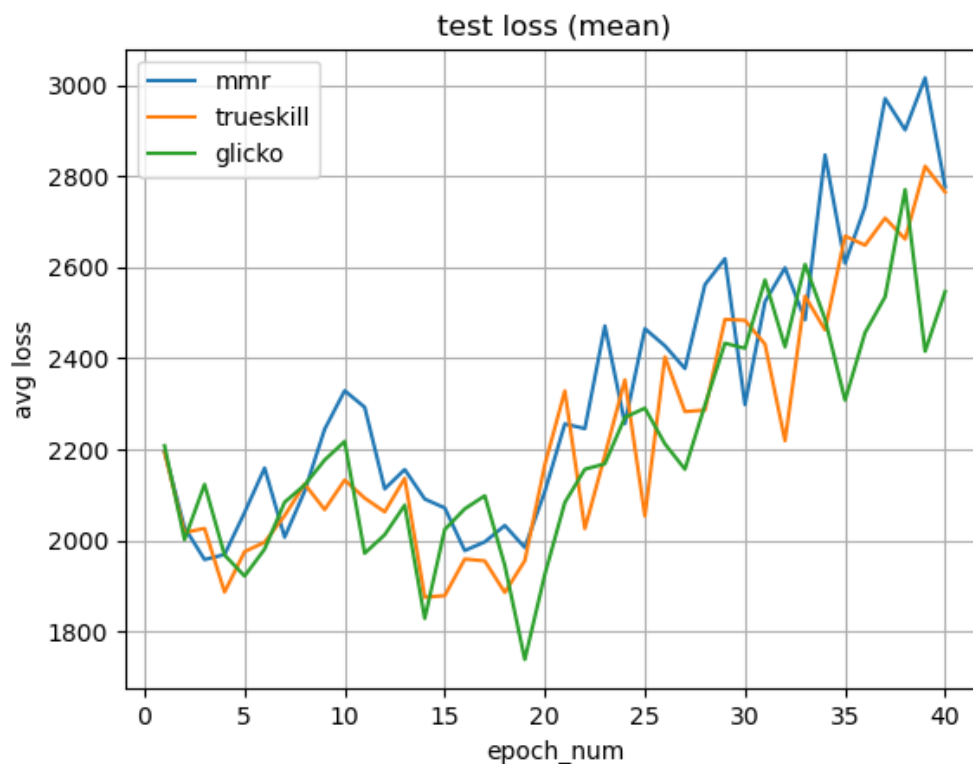


Figure 23. The loss of multilayer perceptron on test data

In Figure 23, this graph shows the average loss of all batches on the test data with respect to the training epoch number. Three curves represent the average loss of the models with different ratings in the input. In the graph, we see that the average loss of all three models has a general decreasing trend from epoch number 1 to epoch number 17. The multilayer perceptron with Elo-MMR rating as input has a low average loss at epoch number 18. The multilayer perceptron with Glicko rating as input has a low average loss at epoch number 18. The multilayer perceptron with TrueSkill rating as

input has low average loss at epoch number 17. After epoch number 18, the average loss of all three models increases remarkably which indicates the overfitting. The model with Elo-MMR rating as input has the highest average loss among the other models while the model with Glicko rating as input has the lowest average loss among the other models.

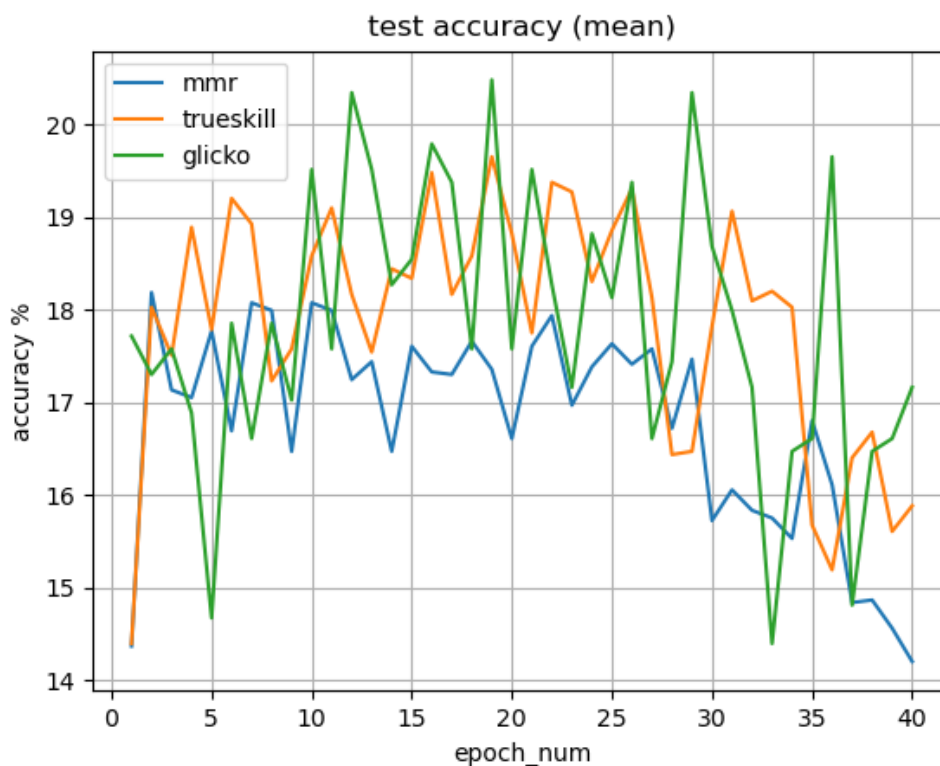


Figure 24. The accuracy of multilayer perceptron on test data

We observe the testing accuracy in Figure 24. We notice that the testing accuracy of all three models keep dropping after the epoch number 17. This is because the models overfit as reflected in Figure 23. The model with Glicko rating reaches the highest test accuracy of 20.4% while the model with Elo-MMR rating has the lowest accuracy among the other models. This is related to the same pattern in the graph of average loss in Figure 23. We can also see that the accuracy of the model with Glicko fluctuates in a larger range than that with Elo-MMR and TrueSkill because the Glicko

rating is dedicated for 2 player games while Elo-MMR and TrueSkill ratings are dedicated for multiplayer games.

5.2.1.2 Betting simulation

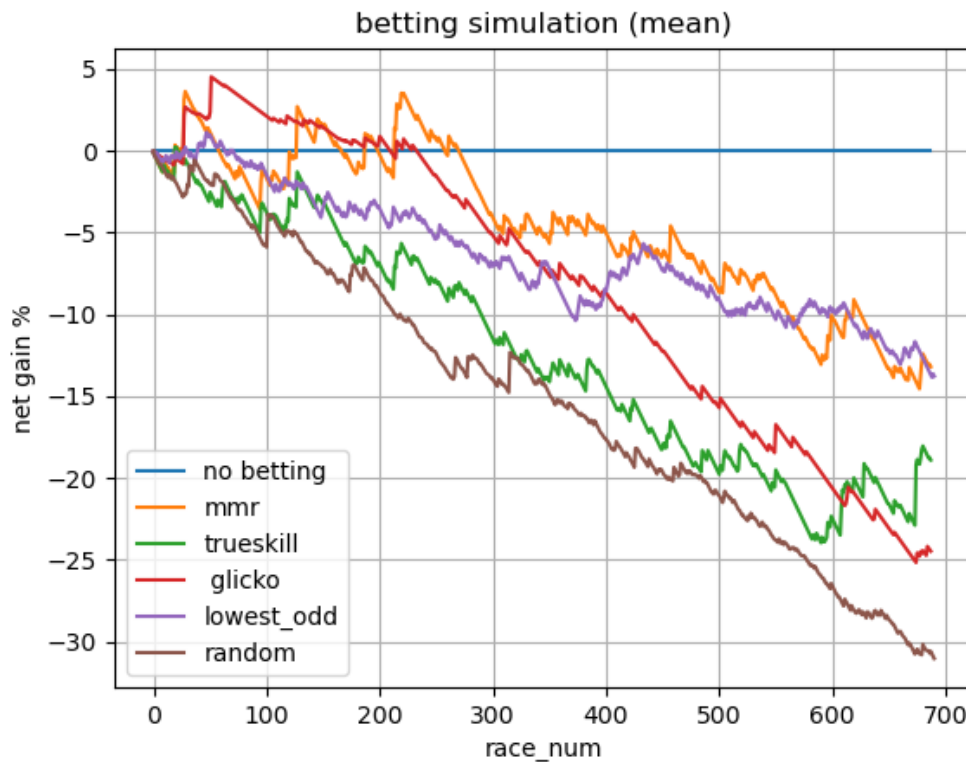


Figure 25. The betting simulation of multilayer perceptron on test data
 We use the prediction of the models to for betting and we show the result in Figure 25. In our betting simulation, all three models perform better than random betting and the model with Elo-MMR rating has similar performance as the lowest odd betting which reflects the public intelligence. This means that the Elo-MMR rating has comparable effect as the winning odds in betting guidance. However, none of the models can give us a positive net gain.

5.2.2 Transformer classification model without ratings

5.2.2.1 Accuracy

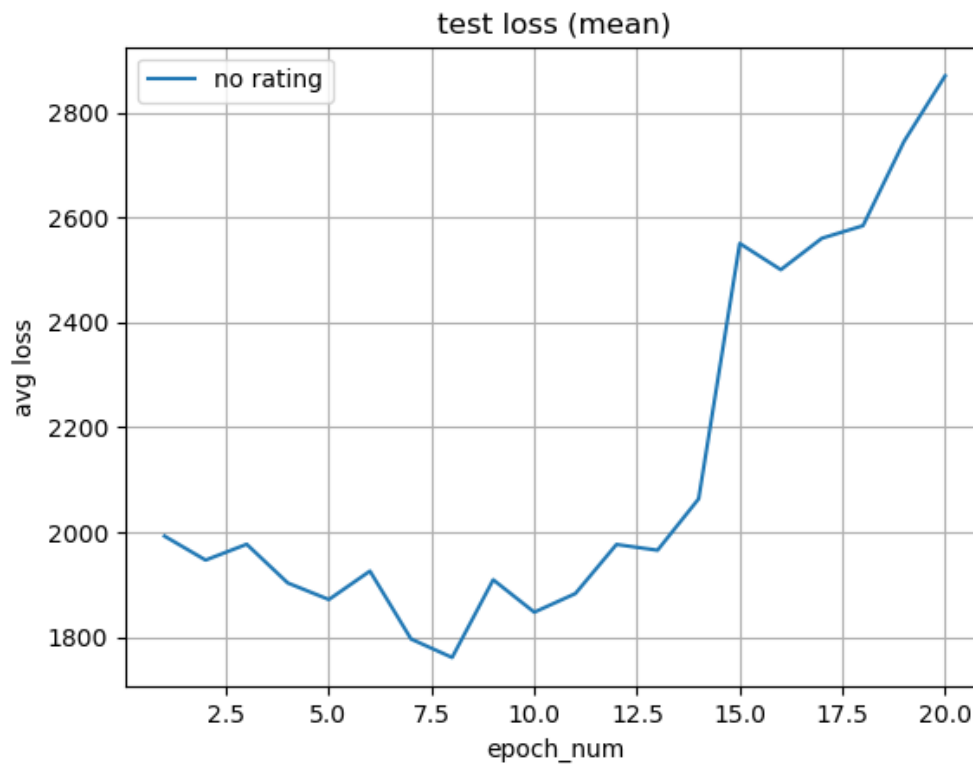


Figure 26. The loss of transformer on test data without rating

In Figure 26, this graph shows the average loss of all batches on the test data with respect to the training epoch number. In the graph, we see that the average loss of this model has a general decreasing trend from the start to epoch number 8. After epoch number 8, the average loss of the model increases remarkably which indicates the overfitting. We observe that this model reaches the converges earlier than models of multilayer perceptron. It is because the transformer classification model is more complex than the multilayer perceptron and it learns faster.

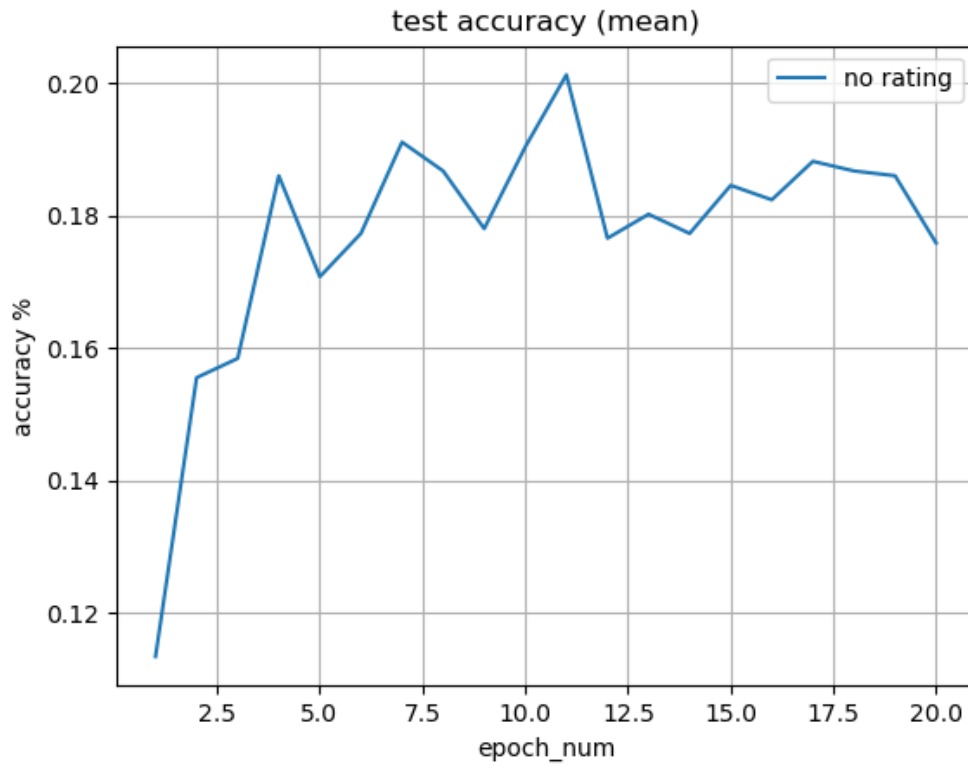


Figure 27. The accuracy of transformer on test data without rating

We examine the testing accuracy in Figure 27. We see that the testing accuracy of this model is in range from 17% to 20% for epoch number larger than 3. The reason for considering the test accuracy after epoch number 3 is that the model is learning and its average loss on test data has not reached the minimum before epoch number 3. From the implication of the average loss in Figure 26, the best performance of this model is having 19.2% at epoch number 6.

5.2.2.2 Betting simulation

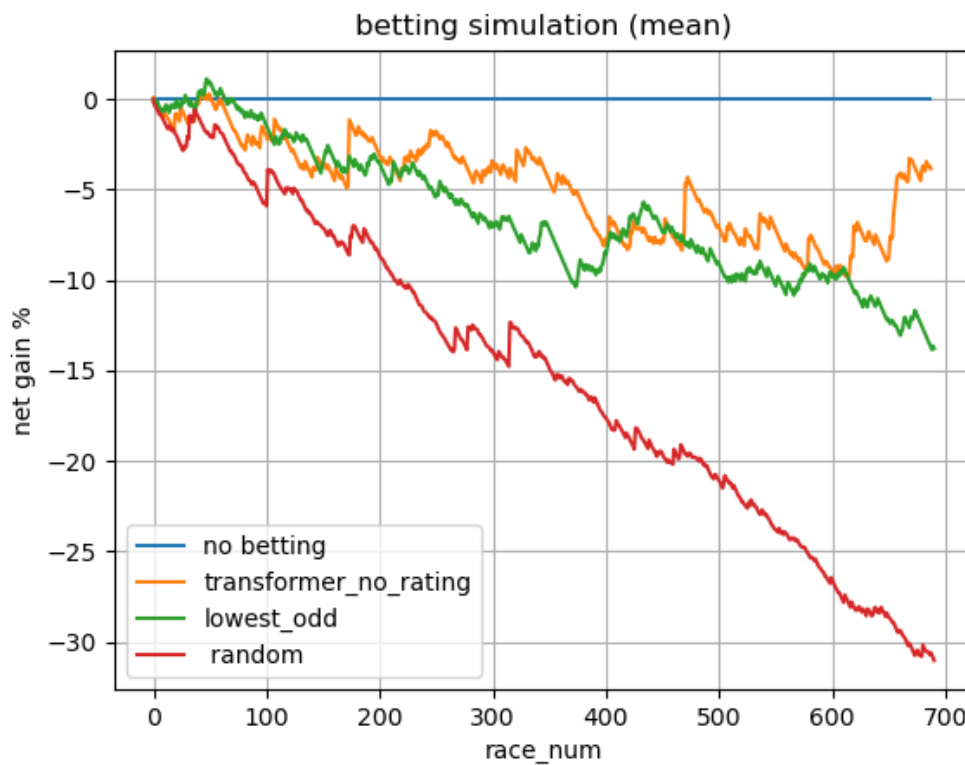


Figure 28. The betting simulation of transformer on test data without rating

We use the prediction of the transformer model to guide us bet on the test data. Figure 28 reveals the performance of this model in profit making aspect. The net gain is -4% after betting on all 688 races in our test data. The performance of this model in betting is better than the multilayer perceptron which has -13% as the highest net gain with Elo-MMR rating included in the input.

5.2.3 Transformer classification model with ratings

5.2.3.1 Accuracy

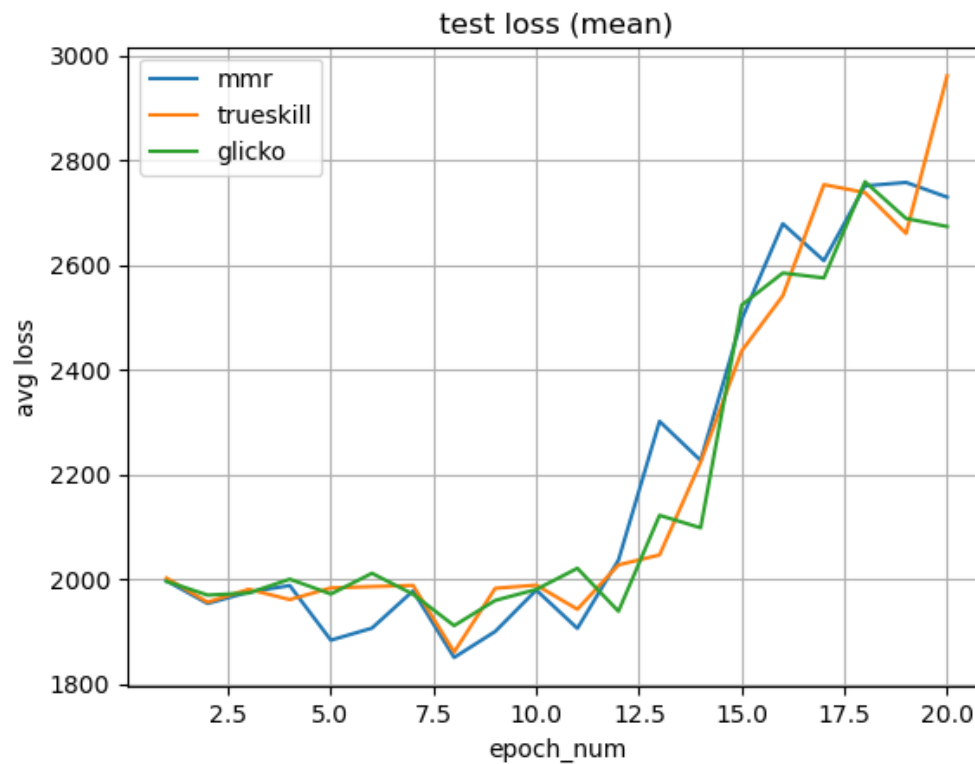


Figure 29. The loss of transformer on test data with rating

Figure 29 shows the average loss of transformer models with different ratings involved in the input. All models overfit after the epoch number 8 because their average loss on test data keep increasing after the epoch number 8. When it is compared to the graph for models using multilayer perceptron, we see that the effect of using different ratings is not significant here because the differences of average loss between the transformer models that use different ratings is smaller than that of the multilayer perceptron models.

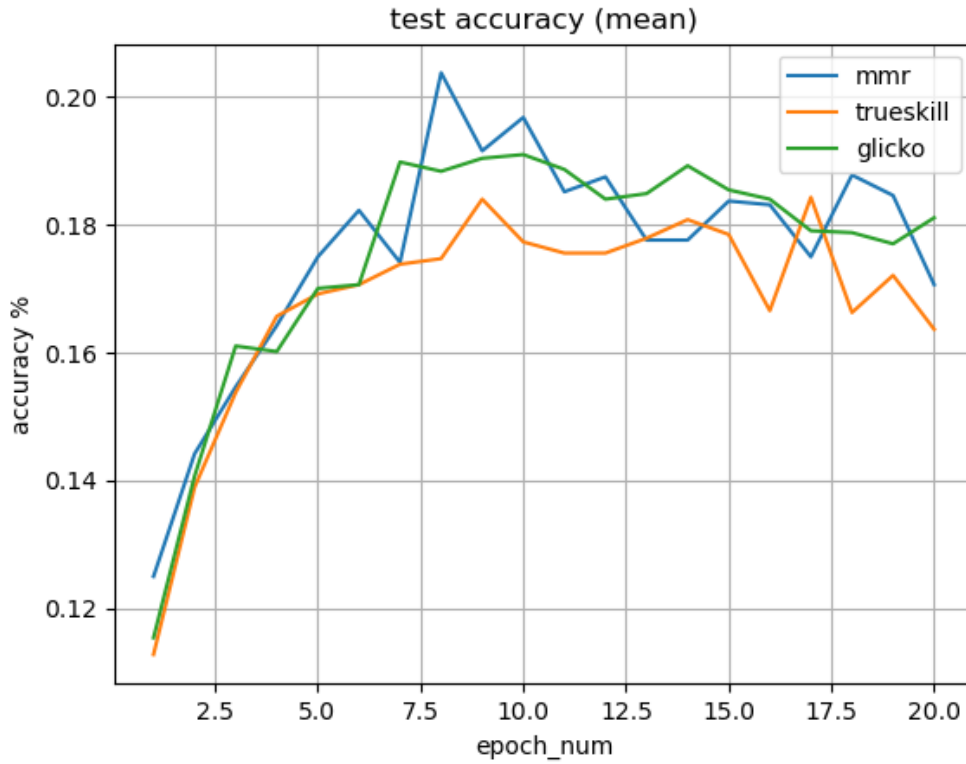


Figure 30. The accuracy of transformer on test data with rating

As Figure 29 indicates overfitting after the epoch number 8, we focus on the test accuracy before the epoch number 8. We notice that the test accuracy increases consistently from the start. The transformer model with Elo-MMR rating included has the highest test accuracy of 21.4% among the other models. Comparing to the test accuracy of transformer model without rating in the input, we conclude that including ratings in the transformer model as input slightly increases the test accuracy. Comparing to the test accuracy of multilayer perceptron models, we conclude that using transformer model slightly increases the test accuracy and narrows down its confidence level because the fluctuation of it is small as shown in Figure 30.

5.2.3.2 Betting simulation

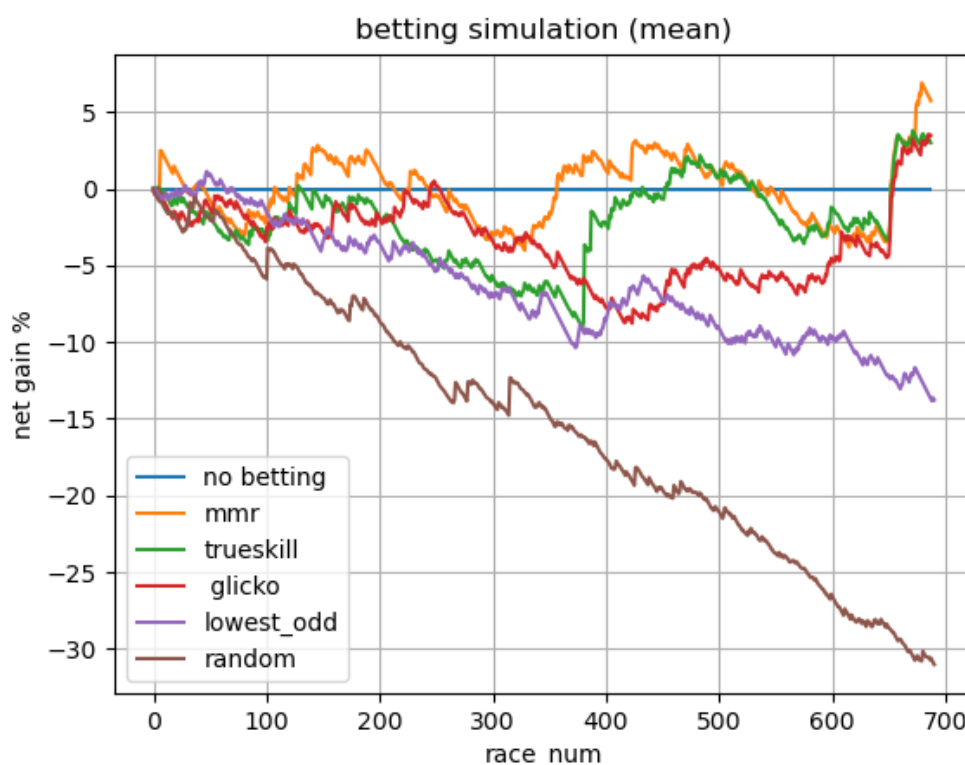


Figure 31. The betting simulation of transformer on test data with rating

When following the predictions of these transformer models in betting, we obtain a satisfactory result as the models give us a positive net gain of 3% to 6% after betting on 688 races in the test data as shown in Figure 31. The transformer models with rating do have a better performance than that of the transformer model without rating and the multilayer perceptron models with rating. Also, we find that the change of net gain is confined to a smaller interval throughout the betting simulation when using transformer model with the Elo-MMR rating.

Chapter 6

Conclusion and Future work

6.1 Conclusion

This project aims at understanding the betting odds in horse racing by analyzing the impact of using both the rating systems and transformer architecture on the accuracy and profit-making aspects of horse racing prediction. From previous studies, the winning odds in the feature list have the effect of enhancing the accuracy and net gain [8][9]. We exclude the winning odds from the feature list this time and attempt to resemble the effect of the winning odds by combining the performance judgement and natural language processing techniques. We contrast the differences in performance by experimenting three models which are multilayer perceptron with ratings, transformer without ratings and transformer with rating. We discover that the best case of our models is the transformer with Elo-MMR ratings which has the highest test accuracy of 21.4% and gives a positive net gain of 6% in betting simulation of the test data. This shows that the combination of ratings and transformer architecture has similar influences on the horse racing prediction. Besides, using transformer architecture in horse racing context can reach the optimal performance with fewer epoch number than that in using traditional deep neural network.

6.2 Future Work

In the next term, we will modify our model into a sequence-to-sequence framework which can predicts the places of all horses in a single race. Then, we will utilize the results in addition to the race and horse features to calculate the winning odds of horses. The second direction is the interpretation of the relationships of features from our neural network models.

References

- [1] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, Perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
- [2] E. Alpaydin, "What Is Machine Learning?," in *Introduction to machine learning*, Cambridge, MA: The MIT Press, 2014.
- [3] P. Cunningham, M. Cord, and S. J. Delany, "Supervised learning," *Machine Learning Techniques for Multimedia*, pp. 21–49.
- [4] S. Becker, "Unsupervised learning procedures for neural networks," *International Journal of Neural Systems*, vol. 02, no. 01n02, pp. 17–33, 1991.
- [5] El Naqa and M. J. Murphy, "What is machine learning?," *Machine Learning in Radiation Oncology*, pp. 3–11, 2015.
- [6] Prieto, M. Atencia, and F. Sandoval, "Advances in artificial neural networks and machine learning," *Neurocomputing*, vol. 121, pp. 1–4, 2013.
- [7] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz, J. Davison, S. Shleifer, P. von Platen, C. Ma, Y. Jernite, J. Plu, C. Xu, T. Le Scao, S. Gugger, M. Drame, Q. Lhoest, and A. Rush, "Transformers: State-of-the-art natural language processing," *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 2020.
- [8] T. T. Cheng and M. H. Lau, "Predicting Horse Racing Result using TensorFlow," Department of Computer Science and Engineering, Hong Kong, 2017.
- [9] Y. Liu and Z. Wang, "Predicting Horse Racing Result with Machine Learning," Department of Computer Science and Engineering, Hong Kong, 2018.
- [10] Y. Wong, "Horse Racing Prediction using Deep Probabilistic Programming with Python and PyTorch (Uber Pyro)," Department of Computer Science and Engineering, 2018.
- [11] "RACING IN HONG KONG," *Hong Kong Racing 101*, Sep-2016. [Online]. Available: https://entertainment.hkjc.com/entertainment/common/chinese/images/more-about-racing/racing-101/Racing-101_201509.pdf. [Accessed: 13-Nov-2021].
- [12] "The Hong Kong jockey club," *About HKJC - The Hong Kong jockey club*. [Online]. Available: <https://corporate.hkjc.com/corporate/english/index.aspx>. [Accessed: 13-Nov-2021].

- [13] R. H. Thaler and W. T. Ziemba, "Anomalies: Parimutuel betting markets: Racetracks and lotteries," *Journal of Economic Perspectives*, vol. 2, no. 2, pp. 161–174, 1988.
- [14] N. Silverman and M. Suchard, "Predicting horse race winners through a regularized conditional logistic regression with frailty," *The Journal of Prediction Markets*, vol. 7, no. 1, pp. 43–52, 2013.
- [15] "Investments and Dividends," *The Hong Kong Jockey Club*. [Online]. Available: <https://special.hkjc.com/infomenu/en/info/investments.asp>. [Accessed: 14-Nov-2021].
- [16] "Betting Guide," *Pari-mutuel local pools - beginners guide - betting entertainment - the hong kong jockey club*. [Online]. Available: https://special.hkjc.com/racing/info/en/betting/guide_qualifications_pari.asp. [Accessed: 13-Nov-2021].
- [17] C. Yau, "Hong Kong jockey club posts record HK\$280 billion revenue despite pandemic," *South China Morning Post*, 31-Aug-2021. [Online]. Available: <https://www.scmp.com/news/hong-kong/hong-kong-economy/article/3147062/hong-kong-jockey-club-posts-record-revenues-hk280>. [Accessed: 13-Nov-2021].
- [18] Davoodi, Elnaz, and Ali Reza Khanteymoori. "Horse racing prediction using artificial neural networks." *Recent Advances in Neural Networks, Fuzzy Systems & Evolutionary Computing* 2010 (2010): 155-160.
- [19] R. N. Bolton and R. G. Chapman, "Searching for positive returns at the track: A multinomial logit model for handicapping horse races," *Management Science*, vol. 32, no. 8, pp. 1040–1060, 1986.
- [20] S. Lessmann, M.-C. Sung, and J. E. V. Johnson, "Alternative methods of predicting competitive events: An application in Horserace Betting Markets," *International Journal of Forecasting*, vol. 26, no. 3, pp. 518–536, 2010.
- [21] W.-C. Chung, C.-Y. Chang, and C.-C. Ko, "A SVM-based committee machine for prediction of Hong Kong horse racing," *2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media)*, 2017.
- [22] GLICKO (Glickman, Mark E. "The glicko system." *Boston University* 16 (1995): 16-17)
- [23] 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.
- [24] Ebtekar and P. Liu, "Elo-MMR: A rating system for massive multiplayer competitions," *Proceedings of the Web Conference 2021*, 2021.

- [25] 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.
- [26] A. Lindholm, "Health aspects of horse production including training with special reference to Nordic conditions," *Livestock Production Science*, vol. 40, no. 1, pp. 73–76, 1994.
- [27] L. Beretta and A. Santaniello, "Nearest neighbor imputation algorithms: A critical evaluation," *BMC Medical Informatics and Decision Making*, vol. 16, no. S3, 2016.
- [28] 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.
- [29] M. K. Dahouda and I. Joe, "A deep-learned embedding technique for categorical features encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021.
- [30] 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.
- [31] EbTech and P. Liu, "EbTech/Elo-MMR: Skill Estimation Systems for multiplayer competitions," *GitHub*. [Online]. Available: <https://github.com/EbTech/Elo-MMR>. [Accessed: 24-Nov-2021].
- [32] F. Murtagh, "Multilayer perceptrons for classification and regression," *Neurocomputing*, vol. 2, no. 5-6, pp. 183–197, 1991.
- [33] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," *arXiv preprint arXiv:1207.0580*, 2012.