THE CHINESE UNIVERSITY OF HONG KONG

THESIS REPORT (TERM 1)

Horse Racing Prediction using Deep Probabilistic Programming with Python and PyTorch (Uber Pyro)

Author:
Yuk WONG

Supervisor:
Prof. Michael R. LYU

LYU1805
Department of Computer Science and Engineering

November 23, 2018

THE CHINESE UNIVERSITY OF HONG KONG

Abstract

Faculty of Engineering

Department of Computer Science and Engineering

BEng degree in Computer Engineering

Horse Racing Prediction using Deep Probabilistic Programming with Python and PyTorch (Uber Pyro)

by Yuk WONG

Probabilistic programming unifies general purpose programming with probabilistic modeling and enables automated inference given probabilistic model. Horse racing prediction is an inherently probabilistic problem, but relatively few progress has been made using probabilistic programming. In this report, we explored the possibility of applying probabilistic programming for horse racing prediction. We showed that our probabilistic programming model can make accurate prediction of individual horse places. Moreover, through repeated experiments, we show that our models can outperform public intelligence in terms of both accuracy and net gain. Finally, we constructed a betting strategy from the training data and verified its profitability in the long run with testing data.

Acknowledgements

We would like to express gratitude to our supervisor Professor Michael R. Lyu and Mr. Edward Lau for providing guidance during this project. Without their guidance, this project will not have the progress it has now.

Contents

Abstract ... 3
Acknowledgements ... 5
Contents ... 7
List of Figures ... 9
List of Tables ... 10
1 Overview ... 12
1.1 Introduction ... 12
1.2 Background ... 13
1.3 Objective ... 15
2 Methodology ... 17
2.1 Finishing time regression ... 17
2.2 Win/lose binary classification ... 17
2.3 Place prediction ... 17
3 Data Preparation ... 18
3.1 Data Collection ... 18
3.2 Data Description ... 18
4 Data Analysis ... 22
4.1 Horse features ... 23
4.1.1 Origin ... 23
4.1.2 Age ... 25
4.1.3 Color ... 27
4.1.4 Sex ... 29
4.2 Race features ... 31
4.2.1 Draw ... 31
4.2.2 Old place ... 33
4.3 Additional features ... 35
5 Data Preprocessing ... 39
5.1 Real Value Data ... 39
5.2 Categorical Data ... 39
6 Model Architecture ... 41
6.1 Stochastic Variational Inference ... 41
6.2 Variational distribution ... 42

6.3 Evidence Lower Bound 42
6.4 Deep Bayesian Neural Network 43
7 Results and Discussion 46
7.1 Results 46
7.2 Discussion 50
8 Conclusion and Future Work 52
8.1 Conclusion 52
8.2 Future Work 52
Bibliography 54

List of Figures

Figure 1 Origin distribution of winning horse 23
Figure 2 Condition Origin distribution of winning horse 24
Figure 3 Age distribution of winning horse 25
Figure 4 Conditional Age distribution of winning horse 26
Figure 5 Color distribution of winning horse 27
Figure 6 Conditional Color distribution of winning horse 28
Figure 7 Sex distribution of winning horse 29
Figure 8 Conditional Sex distribution of winning horse 30
Figure 9 Draw distribution of winning horse 31
Figure 10 Conditional Draw distribution of winning horse 32
Figure 11 Old place distribution of winning horse 33
Figure 12 Conditional Old place distribution of winning horse 34
Figure 13 Deep Bayesian neural network with final sampling layer 44

List of Tables

Table 1 Type of bets in Single-race Pool....14
Table 2 Type of bets in Multi-race Pool....14
Table 3 Types of bets in Jackpot Pool....14
Table 4 Race features from HKJC website....19
Table 5 Extracted features....20
Table 6 Horse features from HKJC website....20
Table 7 Other features included in model input....35
Table 8 Features used in different model input....37
Table 9 Testing performance of different models....46
Table 10 Training Accuracy of model A across different classes....48
Table 11 Training Accuracy of model A + Odds across different classes....48
Table 12 Training Accuracy of model B across different classes....48
Table 13 Training Accuracy of model B + Odds across different classes....48
Table 14 Training Accuracy of model C across different classes....48
Table 15 Training Accuracy of model C + Odds across different classes....48
Table 16 Testing Accuracy of model A across different classes....49
Table 17 Testing Accuracy of model A + Odds across different classes....49
Table 18 Testing Accuracy of model B across different classes....49
Table 19 Testing Accuracy of model B + Odds across different classes....49
Table 20 Testing Accuracy of model C across different classes....49
Table 21 Testing Accuracy of model C + Odds across different classes....49
Table 22 Model B performance with the constructed strategy....50

1 Overview

The goal of this project is to predict horse racing result with deep probabilistic programming and to search for positive return at the field. This report details the work done in the first semester and this chapter overviews the project and introduces the background information, methodology, objective, data preparation,

1.1 Introduction

Probabilistic programming unifies general purpose programming with probabilistic modeling and enables automated inference given probabilistic model. It is mainly applied for making decisions under the face of uncertainty, and it has seen success in artificial intelligence [1], robotics [2], and machine learning [3] [4]. It has also been shown to be effective in optimizing for general objective functions [5] [6] [7].

Deep probabilistic programming combines deep neural networks with probabilistic models [8] and remain as flexible and computation efficient as traditional deep learning [9]. This allows for both automatic function approximation and handling of uncertainty, the best of both worlds, without sacrificing performance. In some studies, it has been shown that deep probabilistic programming can perform better than traditional deep learning [10] [11].

Horse racing prediction is an inherently probabilistic problem and applying probabilistic programming to it is largely unexplored with relatively few published works. We attribute the lack of the progress to the complexity in modeling the problem. Indeed, many different approaches can be used to model the horse performance, and equally many is the number of different bets that can be placed in a race. LYU1603 [12] has worked with two different approaches of binary classification on win/lose and logistic regression on horse finishing time, while LYU1703 [13] focuses on logistic regression on horse finishing time. Both of the projects focus on the “win” and “place” bets. Both projects can generate positive returns, albeit under very limited circumstances: 95% confidence threshold for LYU1603 [12] and betting only on class 1 and 2 for LYU1703 [13]. Nonetheless, this gives a positive outlook for this project to generally generate profit for all races.

In this project, due to the overwhelming number of different available bets, we would also focus on the “win” and “place” bets but uses a different approach of multiclass classification. We showed that our probabilistic programming model can make accurate prediction of individual horse places. Moreover, through repeated experiments, we show that our models can outperform public intelligence in terms of both accuracy and net gain. Finally, we constructed a betting strategy from the training data and verified its profitability in the long run with testing data.

1.2 Background

Horse racing, sport of running horses at speed, is one of the oldest of all sports, and its basic concept has undergone virtually no change over the centuries [14].

In Hong Kong, horse racing is not only a sport but also one of the most important entertainment and gambling activity. The non-profit organization Hong Kong Jockey Club holds a legal monopoly over betting on horse racing and provides different types of bet, according to Pari-mutuel betting system. Pari-mutuel betting is a betting system in which the stake of a particular bet type is place together in a pool, and the returns are calculated based on the pool among all winning bets [15]. Dividend will be shared by the number of winning combinations of a particular pool. Winners will share the percentage of pool payout in proportion to their winning stakes.

The following tables taken from [15] show the different betting type.

Single-race Pool:

Single-race Pool	Dividend Qualification
Win	1^{st} in a race
Place	1^{st}, 2^{nd} or 3^{rd} in a race, or 1st or 2^{nd} in a race of 4 to 6 declared starters
Quinella	1^{st} and 2^{nd} 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 1^{st} horse in a race
Tierce	1^{st} , 2^{nd} and 3^{rd} in correct order in a race
Trio	1^{st}, 2^{nd} and 3^{rd} in any order in a race
First 4	1^{st}, 2^{nd} , 3^{rd} and 4^{th} in any order in a race
Quartet	1^{st}, 2^{nd} , 3^{rd} and 4^{th} in correct order in a race

Table 1 Type of bets in Single-race Pool

Multi-race Pool:

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

Table 2 Type of bets in Multi-race Pool

Jackpot Pool:

Jackpot Pool	Dividend Qualification
Double Trio	1^{st}, 2^{nd} and 3^{rd} in any order in each of the two nominated races
Triple Trio	1^{st}, 2^{nd} and 3^{rd} in any order in each of the three nominated races
	Consolation : Select correctly the 1^{st}, 2^{nd} and 3^{rd} horses in any order in the first two Legs of the three nominated races
Six Up	1^{st} or 2^{nd} in each of the six nominated races
	Six Win Bonus :1^{st} in each of the six nominated races

Table 3 Types of bets in Jackpot Pool

1.3 Objective

This project objective is to apply probabilistic programming to horse racing prediction and build a model to predict horse racing results and generate positive profit under all circumstances. Due to the overwhelming number of different available bets, we focus on “win” and “place” bets, which are the simplest bets and efficiently evaluates the model efficiency in predicting horse racing.

2 Methodology

There are many ways to model the horse racing results. In previous studies, regression on finishing time and binary classification on win/lose are mainly studied. In this project, we focus on multi-class classification of place to model horse performance. Then, we will bet on the best horse will the highest predicted first place score.

2.1 Finishing time regression

Regression on finishing time is a simple yet effective way to interpret horse racing results. In this approach, finishing time of each individual horse are predicted and the horses are ranked based on the predicted time. However, it is unreasonable to impose a distribution model over the finishing time and therefore we do not use this approach in this project.

2.2 Win/lose binary classification

Binary classification on win/lose is another straightforward way to predict whether the horse is going to win. However, binary labeling the data of win/lose will result in highly uneven distributed labels with less than 10% of positive data and more than 90% of negative data.

2.3 Place prediction

Directly predicting the place of the horses is more complicated method but give even data to each class. Although this may result in duplicated place within the same race, the score for each place can be used for ranking the horses in a race.

Moreover, score of each place of a horse can be interpreted as the probability of the horse getting each place, which facilitates building a probabilistic model. Therefore, we focus on this approach in this project.

3 Data Preparation

3.1 Data Collection

Many companies sell horse racing data online. One possible way to obtain training data for our model is to purchase from them. However, due to the lack of budget and the questionable authenticity of these data, we decided to collect the data from the official website of Hong Kong Jockey Club.

3.2 Data Description

The horse racing dataset contains racing data from Jan 1 2011 to April 21 2018. Each entry in the data set represent the information of a horse in a race. The dataset contains 71482 records from 5740 races taken place in Hong Kong. The following tables describes the features obtained from HKJC website.

Feature	Description	Types	Values
raceyear	Year of the race	Date	-
racemonth	Month of the race	Date	-
raceday	Day of the race	Date	-
raceid	Unique id of the race	Index	-
location	Location of the race	Categorical	ST, HV
class	Class of the horses	Categorical	Class 1 to 5, Group 1 to 3
distance	Distance of the race	Categorical	1000, 1200, 1400, 1600, 1650, 1800, 2000, 2200, 2400
course	Track used for the race	Categorical	A, A+3, AWT, B, B+2, C, C+3
going	Soil measurement	Categorical	FIRM, GOOD TO FIRM, GOOD, GOOD TO YIELDING, YIELDING, YIELDING TO SOFT, FAST, SLOW, WET FAST, WET SLOW
raceno	Race number in a race day	Categorical	1 to 8
horseno	Number assigned by HKJC to horse	Categorical	1 to 14
horseid	Unique id of horse	Categorical	4086 distinct values
jockeycode	Unique id of jockey	Categorical	164 distinct values
trainercode	Unique id of trainer	Categorical	146 distinct values
draw	Draw of the horse in race	Categorical	1 to 14
actualweight	Weight added to horse	Real value	-
horseweight	Weight of horse itself	Real value	-
winodds	“win” odds of horse	Real value	1 to 99
place	Place of horse in race	Categorical	1 to 14
finishtime	Finishing time of horse	Real value	-

Table 4 Race features from HKJC website

Apart from the obtaining raw data, we also add some features extracted from the data as follows:

Feature	Description	Types	Values
dn	Day or Night	Categorical	D, N
old_place	Place of horse in last race	Categorical	1 to 14
weightdiff	Difference in weight from previous race	Real value	-

Table 5 Extracted features

In addition, we also obtained horse data from HKJC, which contains useful features such as the origin, age, color, and sex of the horse.

Feature	Description	Types	Values
origin	Place of origin	Categorical	D, N
age	Age of horse	Real value	3 to 10
color	Color of horse	Categorical	-
sex	Sex of horse	Categorical	Colt, Filly, Gelding, Horse, Mare, Rig
sire	Father of horse	Categorical	
dam	Mother of horse	Categorical	
dam's sire	Maternal grandfather of horse	Categorical	
horseid	Unique id of horse	Categorical	4086 distinct values

Table 6 Horse features from HKJC website

4 Data Analysis

For our model input, we do not use all the data for input, for example, we believe that the raceyear, raceday, raceid, raceno, horseno should have no effect on the horse performance, therefore they are excluded. The effect of other features is studied in the following section.

4.1 Horse features

4.1.1 Origin

Historically, the best performing horses comes from Britain, Ireland, and the United States, but recently some of the best horses come from Australia and New Zealand [16]. In addition, the guiding principle for breeding winning racehorses has always been best expressed as "breed the best to the best and hope for the best" [14]. Therefore, the origin of the horse may have an impact on the horse performance.

To analyze whether the origin of the horse is really correlated to the winning probability, we have plotted the origin distribution of winning horse.

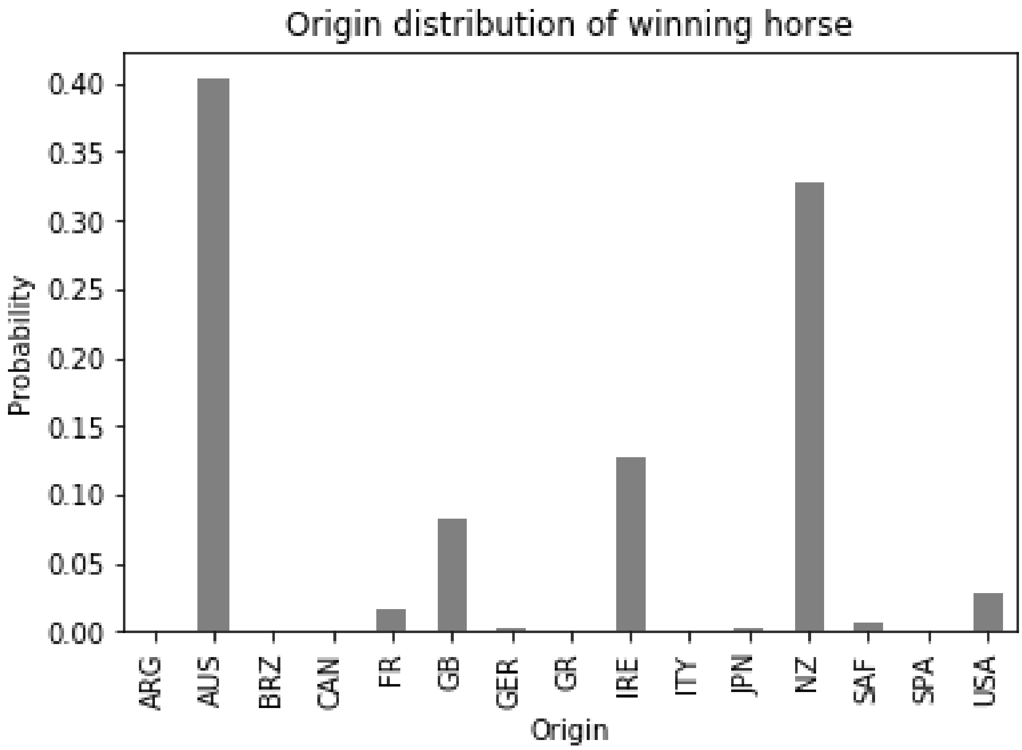


Figure 1 Origin distribution of winning horse

Note that the origin distribution of winning horse alone is not enough to determine the winning probability of horses of different origin, since the distribution is obscured by the origin distribution of all horses. Therefore, we also plot the conditional origin distribution of winning horses.

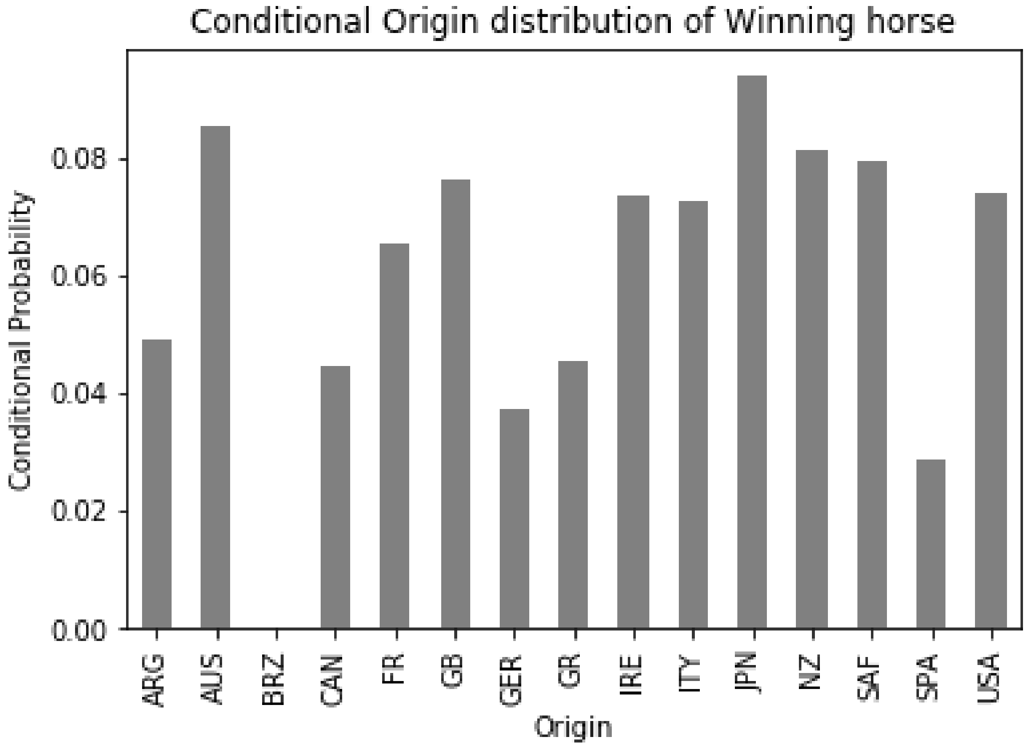


Figure 2 Condition Origin distribution of winning horse

From the figure, it can be inferred that the origin of the horses has an influence over the winning probability with horses from Japan and Australia having the highest winning probability and horses from Brazil and Spain having the lowest winning probability. Therefore, the origin of the horses should be included for the model input.

4.1.2 Age

The racing career of the horse is from age 2 to 10 and retirement is mandatory at age 11. The age of the horse is directly related to its performance. Usually, horses reach their peak performance at age 4 to 6 [17], and start to age subsequently and decrease in performance.

To verify whether the statement above is true, the age distribution and the conditional age distribution of the winning horse is shown below.

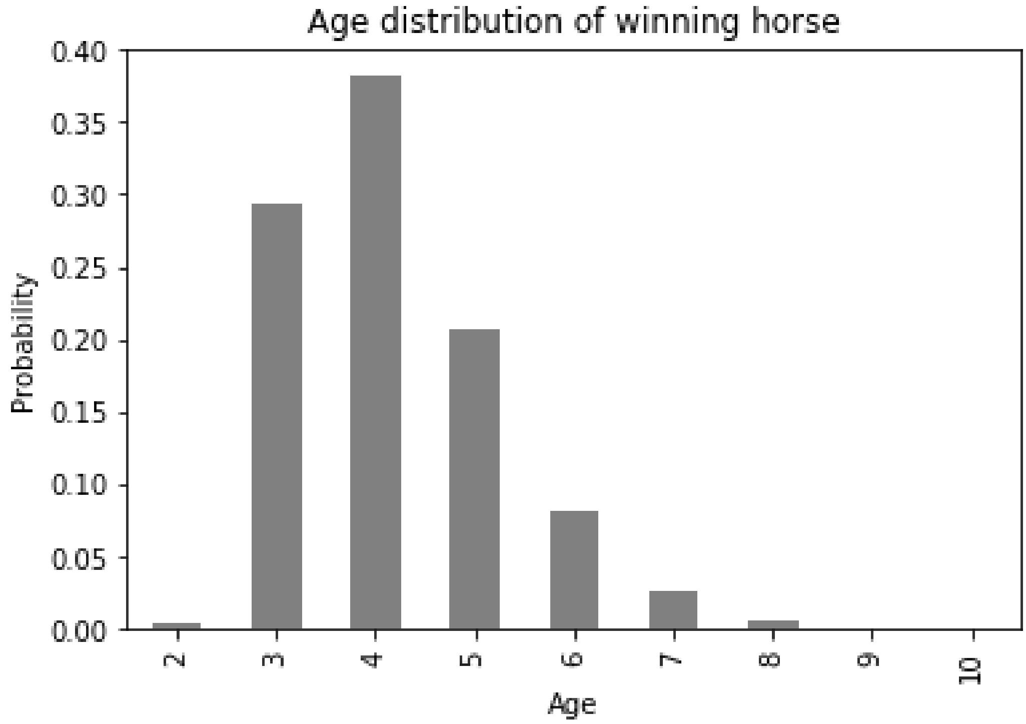


Figure 3 Age distribution of winning horse

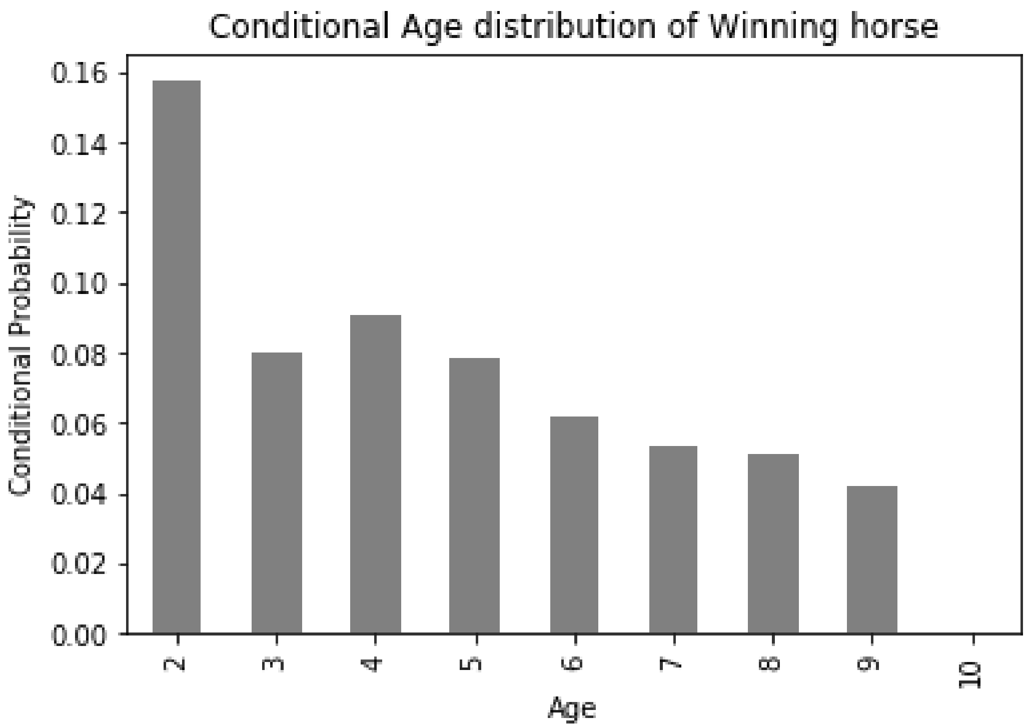


Figure 4 Conditional Age distribution of winning horse

From the figure, it can be inferred that horse's age has a large influence on performance, with age 2 horses having the highest winning probability and age 4 having the second highest winning probability. Therefore, age is an important feature for predicting the winning horse and should be included for the model input.

In addition, it should be noted that age 2 horses are not common and only contribute to a small number of wins, which may because these age 2 horses are prodigies with exceptional performance. Other horses join horse racing at age 3 and takes a year to gain experience and reach peak performance.

4.1.3 Color

It is commonly believed that the color of the horse indicates the horse's performance. In Hong Kong, the major types of colors are Chestnut, Brown, Bay and Grey [17]. To analyze whether color is a factor correlated to winning probability, the color distribution and the conditional color distribution of the winning horse is shown as follows.

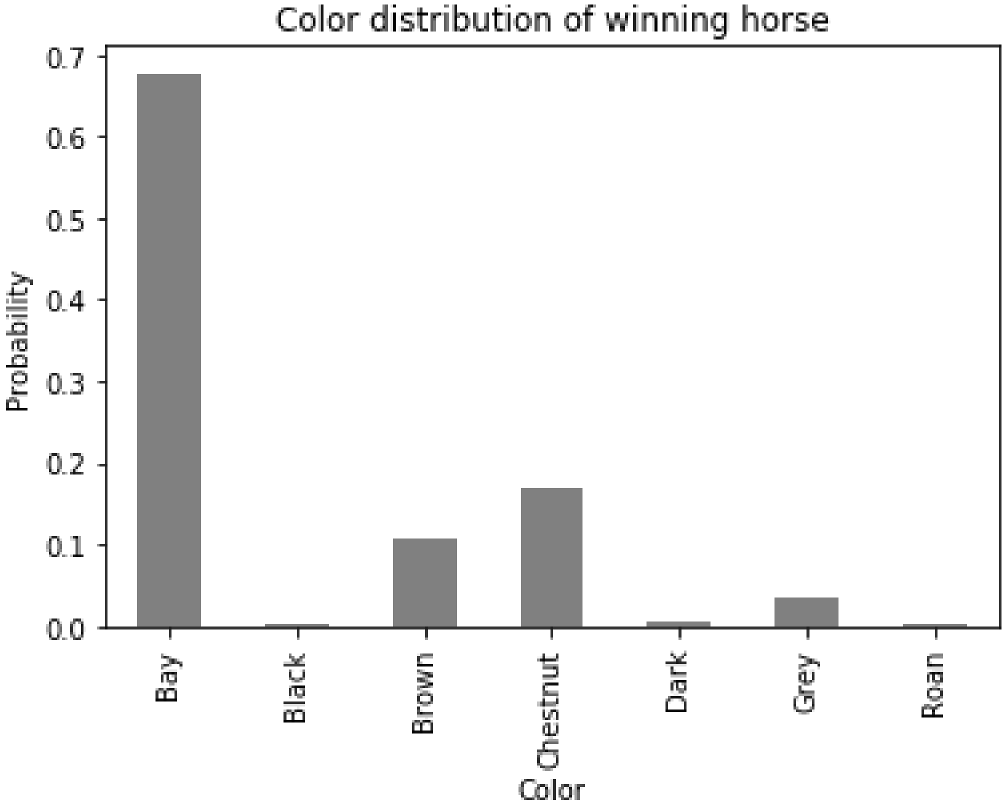


Figure 5 Color distribution of winning horse

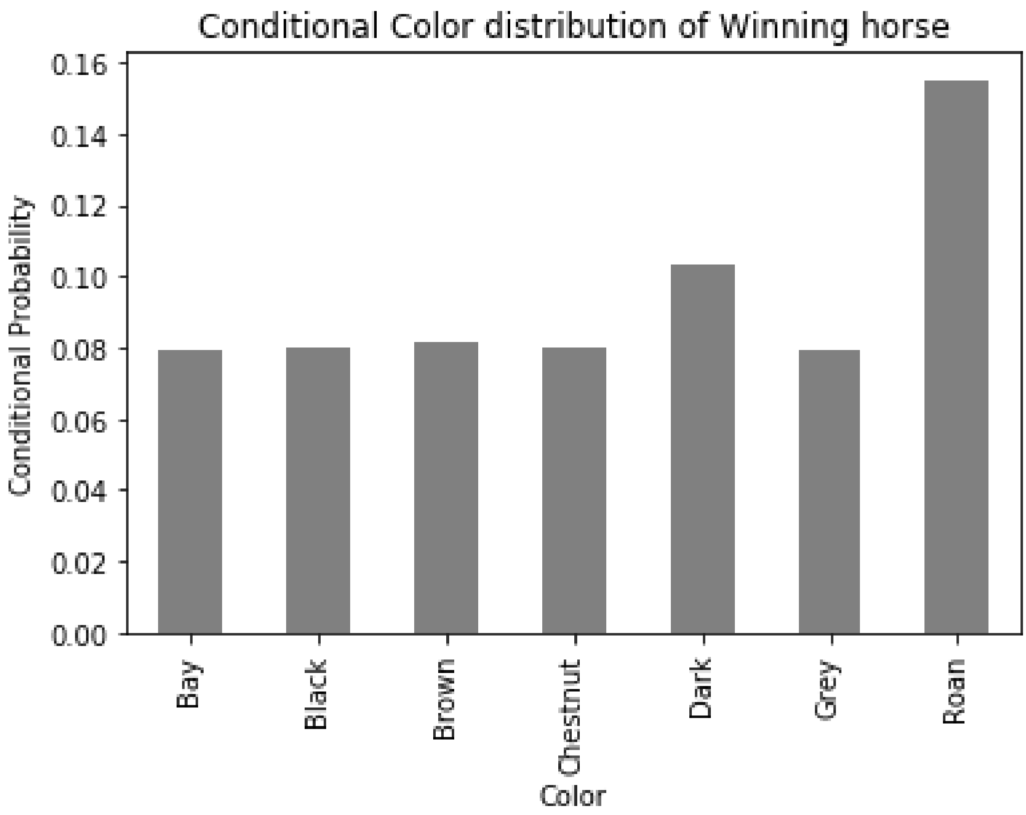


Figure 6 Conditional Color distribution of winning horse

From the figure, it can be inferred that color is correlated to winning probability with horses of dark and roan color being more likely to win while horses of bay, black, brown, and chestnut have similar winning probability. Therefore, color should be included for model input.

4.1.4 Sex

The sex of the horse is mainly classified into Gelding, Colt, or Filly. Over 90% of the runners in Hong Kong are geldings [17]. The different hormones levels of different sex may lead to different performance [18]. To analyze whether sex affects winning performance of the horses, the sex distribution and conditional sex distribution of winning horse is shown below.

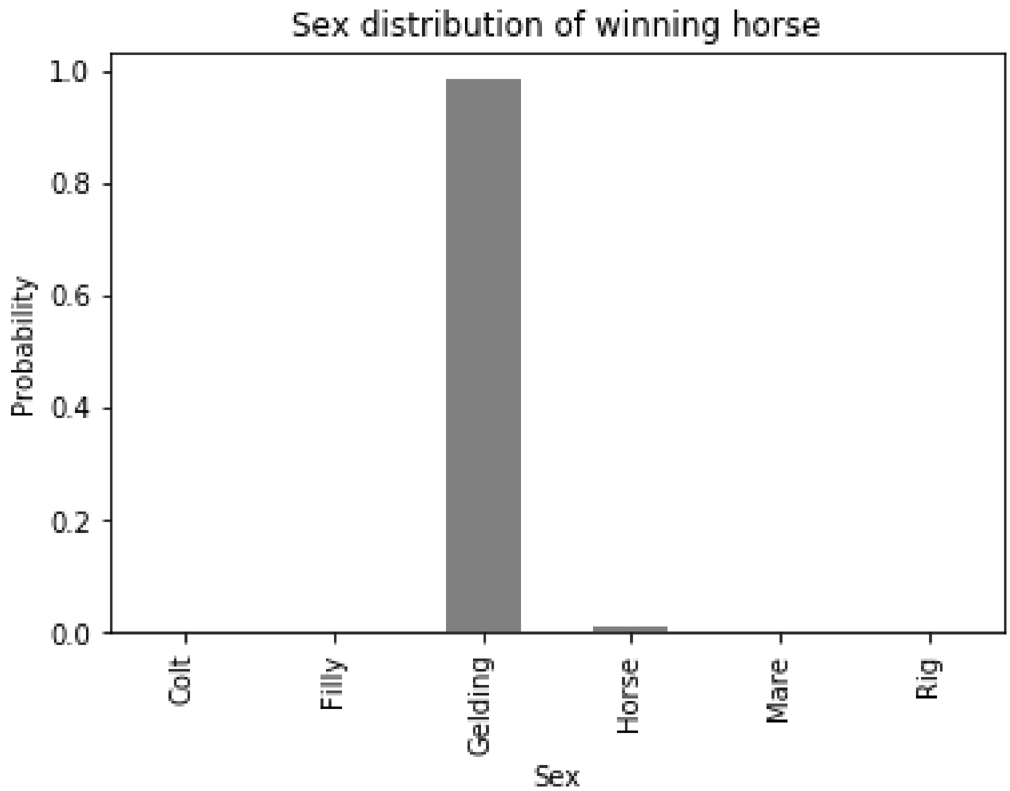


Figure 7 Sex distribution of winning horse

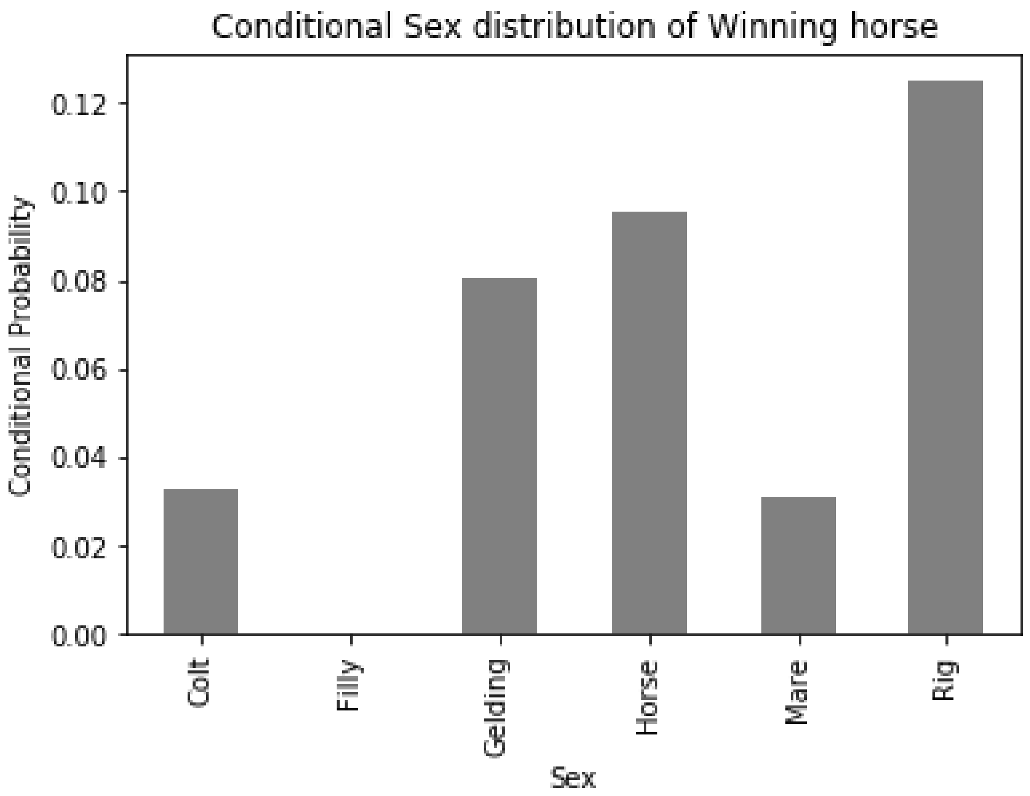


Figure 8 Conditional Sex distribution of winning horse

The analysis of sex distribution of winning horse reveals that sex is also important in determination of winners and should be included for the model input. In general, it can be inferred that male horses (Colt, Gelding, Horse, Rig) has a higher winning probability than female horses (Filly, Mare).

4.2 Race features

4.2.1 Draw

In general, horses starting with an inside draw (smaller draw number) have a competitive advantage, since an inside rail has a shorter distance at turns [19]. However, the distance and the running style of the horse may also impact the influence of draw number. For example, Shatin Turf 1000M Straight has no turns and there is no advantage for having an inside draw. In addition, as there is less damage to the track on the outskirts of the track, horses that start from an outside draw (larger draw number) have a competitive advantage.

To verify whether the above principle is correct, we plot the draw distribution and the conditional draw distribution of winning horse.

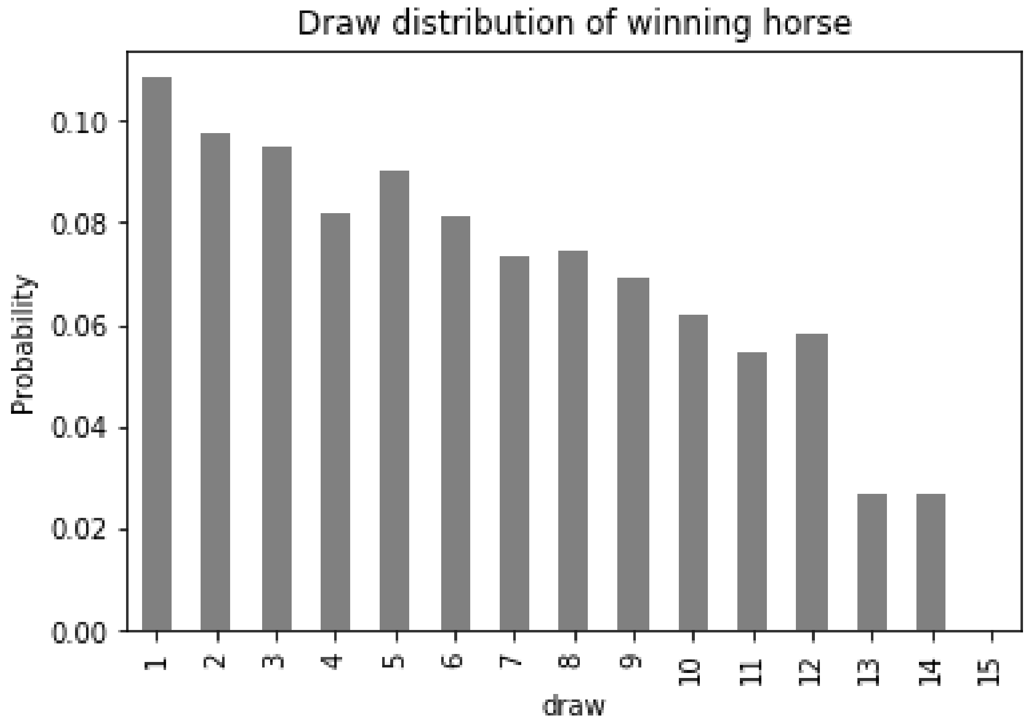


Figure 9 Draw distribution of winning horse

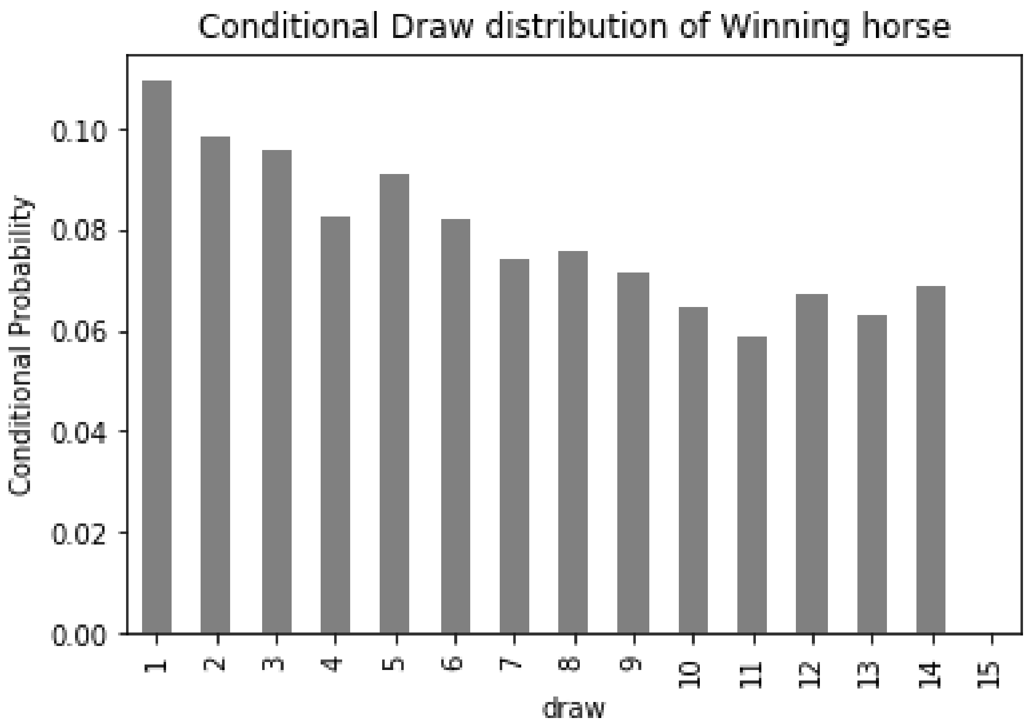


Figure 10 Conditional Draw distribution of winning horse

The above figures indicate that horses with smaller draw number are more likely to win, which supports the general principle of [17]. Therefore, it can be concluded that draw is indeed an important feature and should be included for the model input.

4.2.2 Old place

Apart from the intrinsic characteristic of the horse, the past performance of the horse is also important. Intuitively, a horse with a track record of all first places is more likely to win then a horse with a track record of all last places.

To verify whether our intuition is correct, we plot the old place distribution of winning horse and the conditional old place distribution of winning horse below.

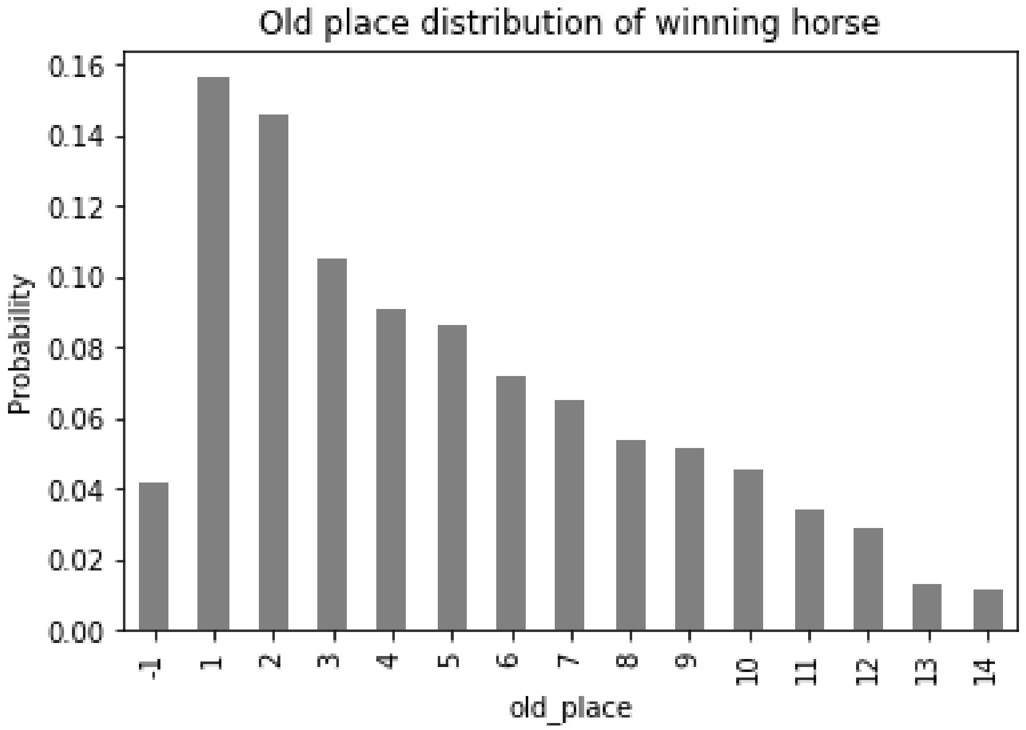


Figure 11 Old place distribution of winning horse

Here, -1 indicates that there is no past records for the horse.

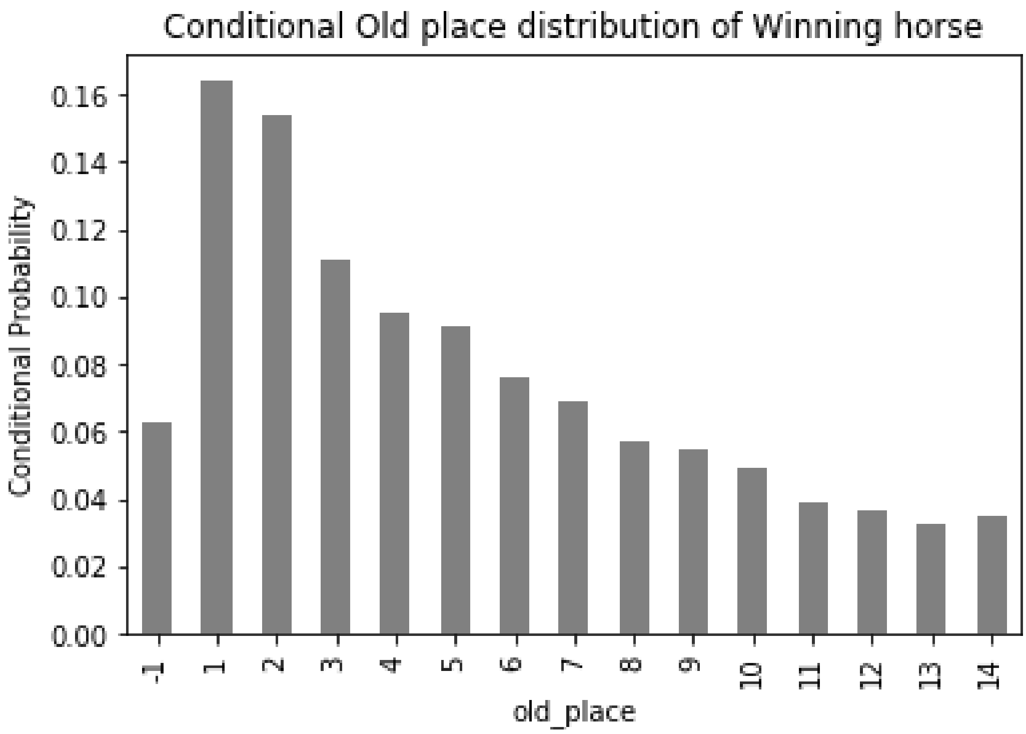


Figure 12 Conditional Old place distribution of winning horse

Here, -1 indicates that there is no past records for the horse.

The data shown has clearly points out that winners will remain winners, and losers will remain losers. Therefore, the old place of the horse is also an important feature for prediction of horse place and should be included for model input.

4.3 Additional features

Although the above analysis maybe useful for features directly correlated to horse performance, some features may only be correlated to horse performance in combination with other features. Also, some of the features may be the same for all horses in the same race. The table below summarize these features and give support for adding these features.

Feature	Reason
racemonth	Month of the race may affect weather, which may in turn affect horse performance.
location	Different locations have different tracks. Shatin tracks are flat while Happy Valley track has ups and downs.
class	Different class have different horse strength, which requires different characteristics for winning.
distance	Long distance requires more endurance, while short distances requires more speed. Also, longer distances involve more turns.
course	Different courses have different characteristics. In a dirt track, kickback may hit horses at the rear of the field.
going	The soil condition affects the way horse runs.
horseid	Different horse behaves differently even if they are similar in other features.
jockeycode	A better jockey will lead to higher chances of winning.
trainercode	Some trainers may perform well on particular track surface.
actualweight	A heavy weight will cause the horse to run slower.
horseweight	For lighter horses, it is harder to carry a heavy weight than heavier horses.
winodds	This reflects public intelligence.
dn	Horses may behave differently during day time and night time.
weightdiff	The weight difference reflects the health condition of the horse.
sire	Horses of same sire should have similar performance.
dam	Horses of same dam should have similar performance.
dam’s sire	Horses of same dam’s sire should have similar performance.

Table 7 Other features included in model input

To investigate the effect of adding additional features, we iteratively add more features and test the accuracy.

First, we believe that all non-entity features should be used for input, i.e., every feature except *horseid*, *trainercode*, *jockeycode, sire, dam, dam's sire* should be used. Entity features are too specific and repeat only a few times inside the data set, and therefore are removed. Thus, we generate the first model, Model A, with an input of 16 features.

Then, we add *trainercode* and *jockeycode* to the input, as there are only 1XX trainers and jockeys, and therefore allow for enough repetition for a model to learn. Then, we generate the second model, Model B, with an input of 18 features.

Finally, we add *horseid, sire, dam, dam's sire* to the input, creating a final model, Model C, with an input of 22 features.

One of the important differences between LYU1603 [12] and LYU1703 [13] is the use of odds data. We believe that while adding odds data to input increases prediction accuracy, odds data may push model away from predicting horse with higher odds which means higher return. To compare whether adding odds data increases prediction accuracy, we have created variant with odds data A + Odds, B + Odds, and C + Odds of model A, B, and C respectively.

The following table shows the features used in each model.

Feature	A	A + Odds	B	B + Odds	C	C + Odds
racemonth	X	X	X	X	X	X
location	X	X	X	X	X	X
class	X	X	X	X	X	X
distance	X	X	X	X	X	X
course	X	X	X	X	X	X
going	X	X	X	X	X	X
horseid					X	X
jockeycode			X	X	X	X
trainercode			X	X	X	X
draw	X	X	X	X	X	X
actualweight	X	X	X	X	X	X
horseweight	X	X	X	X	X	X
winodds		X		X		X
dn	X	X	X	X	X	X
old_place	X	X	X	X	X	X
weightdiff	X	X	X	X	X	X
origin	X	X	X	X	X	X
age	X	X	X	X	X	X
color	X	X	X	X	X	X
sex	X	X	X	X	X	X
sire					X	X
dam					X	X
dam’s sire					X	X

Table 8 Features used in different model input

5 Data Preprocessing

5.1 Real Value Data

We apply normalization on real value data to make training less sensitive to the scale of individual features. We use the z – score normalization to make the data have zero mean and unit variance. To prevent information leakage, we use the mean and variance of the training data for normalization. The data is then normalized according to the following equation:

$$\hat{X} = \frac{X - mean(X)}{std(X)}$$

5.2 Categorical Data

We use one hot encoding to represent categorical data. This approach, while creating a high dimension and memory intensive, represents categorical data in an unbiased way so that every class is equally separated and unrelated. This approach is also the most straight forward way to represent categorical data.

One approach to overcome the high dimensionality and large memory consumption is to train an embedding network for each of column of the data set. However, this requires careful selection of embedding dimension and complicated network design. Since our dataset are still well within the size of available memory, it is not deemed as necessary to use embedding networks.

6 Model Architecture

In traditional neural networks, the weight and bias are considered as parameters with only a single value, i.e., they are point estimates [20]. In Bayesian neural networks, there are uncertainty in the estimation of weight and bias and therefore they are also random variables. In addition, they are not directly observed, thus, they are the latent random variables.

6.1 Stochastic Variational Inference

Suppose the model has observations **x** and latent random variables **z** and parameters θ. It has a joint probability of the form

$$p_\theta(\boldsymbol{x}, \boldsymbol{z}) = p_\theta(\boldsymbol{x}|\boldsymbol{z})p_\theta(\boldsymbol{z})$$

Then, we wish to maximize the log evidence, i.e., we wish to tune parameters θ such that $\log p_\theta(\boldsymbol{x})$ is maximum.

$$\theta_{max} = \operatorname*{argmax}_\theta \log p_\theta(\boldsymbol{x})$$

Where log evidence $\log p_\theta(\boldsymbol{x})$ is calculated by

$$\log p_\theta(\boldsymbol{x}) = log \int d\boldsymbol{z}\, p_\theta(\boldsymbol{x}, \boldsymbol{z})$$

Since we wish to obtain the weight and bias of the neural networks, we also wish to compute the posterior over the latent variables **z**:

$$p_{\theta_{max}}(\boldsymbol{z}|\boldsymbol{x}) = \frac{p_{\theta_{max}}(\boldsymbol{x}, \boldsymbol{z})}{\int d\boldsymbol{z}\, p_{\theta_{max}}(\boldsymbol{x}, \boldsymbol{z})}$$

Variational inference [21] enables us to find θ_{max} and computes the approximation to the posterior $p_{\theta_{max}}(\boldsymbol{z}|\boldsymbol{x})$, by introducing a parameterized distribution as an approximation to the posterior.

6.2 Variational distribution

A parameterized distribution $q_\phi(\boldsymbol{z})$ is introduced to approximate the posterior and enables us to compute the approximation of the posterior over the latent variables **z**. Here, ϕ are known as the variational parameters.

Then, the learning problem can be setup as an optimization problem where we wish to find θ and ϕ so that the variational distribution is close to the exact posterior. To do this we would need to define an approximate objective function for evaluate the “clossness” of the variational and the exact posterior.

6.3 Evidence Lower Bound

A simple derivation from [22], the evidence lower bound, enables us to measure the “clossness” between the variational distribution and the exact posterior:

$$\text{ELBO} \equiv E_{q_\phi(\boldsymbol{z})}[\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) - \log q_\phi(\boldsymbol{z})]$$

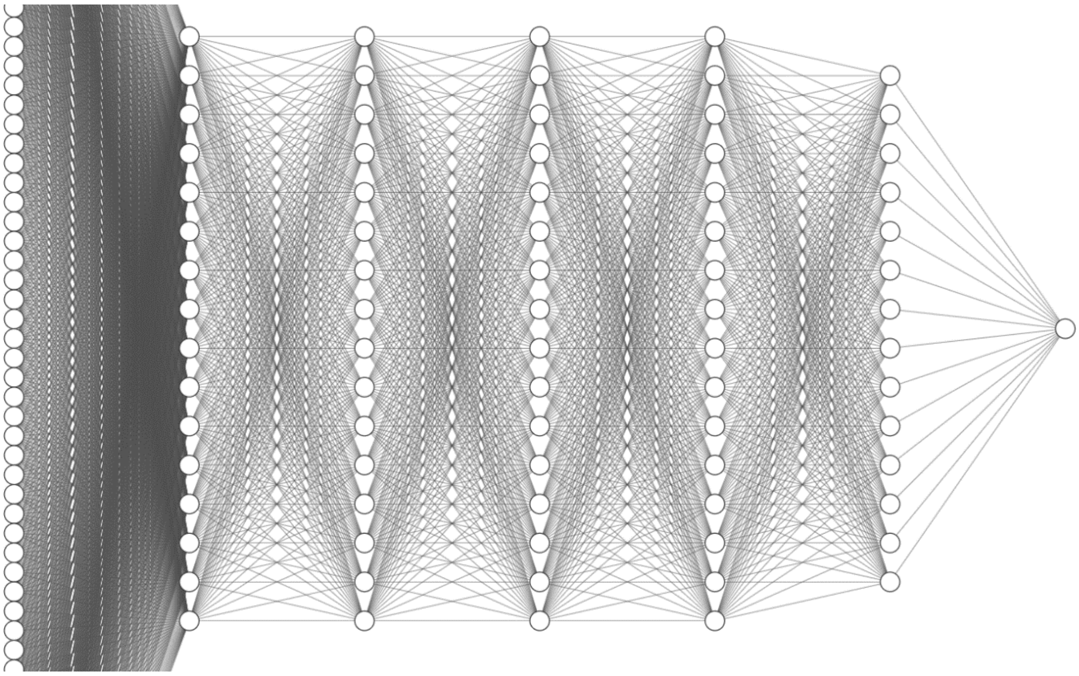
By tuning θ and ϕ to maximize ELBO, we can optimize our variational distribution to better approximate the exact posterior. However, at a first glance, this has no relationship with how good our model truly is. Indeed, it is not very useful our variational distribution can only approximate the posterior of the model we build our self.

Since $\log p_\theta(\boldsymbol{x}, \boldsymbol{z}) \geq$ ELBO, by maximizing ELBO, we also be pushing the log evidence higher. In other words, when we optimize our guide, our model is also improved.

Thankfully, Uber Pyro provides support for doing variational inference, so the above will be done in a few lines of code.

6.4 Deep Bayesian Neural Network

In this project, we use deep Bayesian neural network for predicting the place of each horse. A Bayesian neural network is a neural network with a prior distribution on its weights and biases [23], and extends standard neural networks with posterior inference [24]. The key design of our model is the final sampling of the place from the output of the network as the probability. The model can be broken into 2 parts, a fully connected neural network and an additional sampling layer. The sampling layer takes the output of the output layer as the probabilities of the places and randomly samples a place. For example, suppose the output of the output layer is [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0]. Then, the sampling layer would sample the places with [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0, 0, 0, 0] for places 1 to 14 respectively.



The figure above shows the design of the network with 1 input layer, 4 hidden layers, 1 output layer, and 1 sampling layer. Each of the hidden layers contains 16 neurons. Tanh is used for activation function in input and hidden layer, and SoftMax is used for output layer.

The flow of the model is shown below:

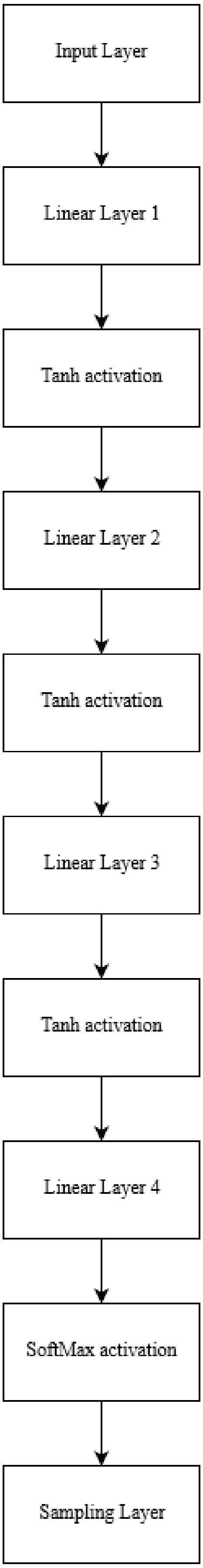


Figure 13 Deep Bayesian neural network with final sampling layer

7 Results and Discussion

7.1 Results

We split our data set into training data and testing data, using data from Jan 1 2011 to Mar 29, 2017 for training and data from Apr 2, 2017 to April 21 2018 for testing. This result in 57334 training data and 10063 testing data.

We have trained our models with 800,000 iterations over the training dataset with Adam optimizer with an initial learning rate of 0.0001. Since the probabilistic model we build outputs a different neural network every time, we sample 100 different set of neural weights and biases from our model and take the average performance as the result.

We use the following criteria to evaluate the different models:

1. Accuracy: accuracy of win and place bets in testing dataset
2. Net gain: overall net gain of win over the testing dataset and ratio of return over bet

The following table shows the performance of different model. The bet based on the lowest odds (public intelligence) is also shown.

Model	*LowestOdds*	*A*	*A+Odds*	*B*	*B+Odds*	*C*	*C+Odds*
$Accuracy_{win}$	0.2614	0.1840	0.2576	0.1798	0.2592	0.1830	0.2634
$Accuracy_{place}$	0.5709	0.4513	0.5695	0.4479	0.5774	0.4551	0.5816
Net gain	-224.90	-184.68	-184.5	-177.45	-164.65	-220.29	-188.06
Return/Bet	-0.2637	-0.2165	-0.2163	-0.2080	-0.2009	-0.2583	-0.2205

Table 9 Testing performance of different models

In terms of accuracy, the models using odds data have similar performance with public intelligence and the models not using odds data have lower performance. In terms of net gain and return over bet, however, all the models perform better than public intelligence.

From these results it can be concluded that using odds data can increase prediction accuracy but have minimal impact on net gain and return over bet. Also, adding more features in addition to odds data can further increase accuracy. Still, none of the models can generate positive profit if betting is done on every race and the best return/bet of -0.2080/-0.2009 is achieved with model B/B + Odds.

LYU1703 [13] suggested that the accuracy varies across different classes, and therefore by betting only on specific classes they can generate a positive profit. Here, we also analyze the performance of the model across different classes for the possibility of constructing a betting strategy to generate positive profit.

To prevent information leakage, we formulate the strategy based on only training data and test the strategy on testing data. The training and testing $Accuracy_{win}$ and $Accuracy_{place}$ of different models across different classes are shown on the next page.

Table 10 Training Accuracy of model A across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
Accuracy$_{win}$	0.1764	0.1520	0.1858	0.1764	0.1576	0.1949	0.2267	0.2288
Accuracy$_{place}$	0.4403	0.4089	0.4554	0.4407	0.4019	0.5050	0.5133	0.4669
Net gain	-35.04	-172.91	-304.85	-398.78	-178.59	-12.85	0.38	-4.94
Return/Bet	-0.2558	-0.3886	-0.2007	-0.2308	-0.2551	-0.1428	0.0088	-0.0659

Table 11 Training Accuracy of model A + Odds across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
Accuracy$_{win}$	0.2972	0.3351	0.2950	0.2655	0.2643	0.3433	0.3474	0.3149
Accuracy$_{place}$	0.6364	0.6359	0.5947	0.5736	0.5309	0.7096	0.6886	0.6145
Net gain	-16.33	-35.37	-169.29	-250.75	-44.98	-11.20	-1.15	-10.27
Return/Bet	-0.1192	-0.0795	-0.1114	-0.1451	-0.0643	-0.1244	-0.0268	-0.1369

Table 12 Training Accuracy of model B across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
Accuracy$_{win}$	0.2536	0.2522	0.2254	0.2120	0.2168	0.2558	0.2302	0.3349
Accuracy$_{place}$	0.5199	0.5456	0.4987	0.5027	0.4476	0.5626	0.5721	0.5952
Net gain	1.69	-54.45	-254.67	-304.50	-19.43	-18.80	-8.32	10.86
Return/Bet	0.0123	-0.1224	-0.1677	-0.1762	-0.0278	-0.2089	-0.1935	0.1448

Table 13 Training Accuracy of model B + Odds across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
Accuracy$_{win}$	0.3049	0.3232	0.2902	0.2655	0.2583	0.3176	0.2826	0.3084
Accuracy$_{place}$	0.6431	0.6450	0.5987	0.5746	0.5264	0.6916	0.6570	0.6012
Net gain	-6.70	-38.07	-182.83	-215.37	-36.46	-17.93	-7.38	-11.07
Return/Bet	-0.0489	-0.0856	-0.1204	-0.1246	-0.0521	-0.1992	-0.1717	-0.1476

Table 14 Training Accuracy of model C across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
Accuracy$_{win}$	0.2281	0.2282	0.2173	0.2019	0.2123	0.2141	0.2147	0.2325
Accuracy$_{place}$	0.4964	0.5278	0.5045	0.4972	0.4600	0.5163	0.5602	0.5140
Net gain	-11.00	-60.55	-121.64	-198.46	-3.39	4.26	2.17	-9.08
Return/Bet	-0.0803	-0.1361	-0.0801	-0.1148	-0.0048	0.0473	0.0504	-0.121

Table 15 Training Accuracy of model C + Odds across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
Accuracy$_{win}$	0.3095	0.3348	0.3079	0.2747	0.2722	0.3291	0.3333	0.3135
Accuracy$_{place}$	0.6362	0.6476	0.6114	0.5855	0.5409	0.7004	0.6874	0.6139
Net gain	-11.94	-38.41	-145.66	-223.08	-31.56	-18.69	-4.14	-12.37
Return/Bet	-0.0871	-0.0863	-0.0959	-0.1291	-0.0451	-0.2076	-0.0963	-0.165

Table 16 Testing Accuracy of model A across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
$Accuracy_{win}$	0.2529	0.2221	0.1875	0.1858	0.1343	0.2367	0.1333	0.1458
$Accuracy_{place}$	0.6414	0.4958	0.4236	0.4732	0.3767	0.7050	0.3789	0.4925
Net gain	7.00	-32.02	-61.65	-46.99	-41.44	-4.35	-4.53	-0.71
Return/Bet	0.4997	-0.3558	-0.2156	-0.1487	-0.3635	-0.3622	-0.5038	-0.0588

Table 17 Testing Accuracy of model A + Odds across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
$Accuracy_{win}$	0.2900	0.3817	0.2684	0.2240	0.2107	0.3392	0.3333	0.2200
$Accuracy_{place}$	0.7414	0.6678	0.5819	0.5535	0.4418	0.8342	0.4611	0.7908
Net gain	-2.54	-6.14	-49.61	-89.77	-30.66	1.03	-1.57	-5.25
Return/Bet	-0.1812	-0.0682	-0.1734	-0.2841	-0.2689	0.0856	-0.1744	-0.4378

Table 18 Testing Accuracy of model B across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
$Accuracy_{win}$	0.2771	0.2756	0.1741	0.1552	0.1407	0.2142	0.4311	0.2825
$Accuracy_{place}$	0.4979	0.5372	0.4566	0.4191	0.3686	0.7867	0.5489	0.6117
Net gain	2.45	-15.57	-59.41	-81.17	-31.70	-1.64	1.69	7.89
Return/Bet	0.1753	-0.1730	-0.2077	-0.2569	-0.2780	-0.1363	0.1879	0.6573

Table 19 Testing Accuracy of model B + Odds across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
$Accuracy_{win}$	0.2271	0.3594	0.2796	0.2293	0.2074	0.3217	0.3311	0.2242
$Accuracy_{place}$	0.6907	0.6730	0.5894	0.5620	0.4556	0.8267	0.4489	0.7575
Net gain	-6.18	-5.66	-40.18	-79.31	-27.83	0.34	-0.94	-4.89
Return/Bet	-0.4413	-0.0629	-0.1405	-0.2510	-0.2441	0.0287	-0.1047	-0.4078

Table 20 Testing Accuracy of model C across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
$Accuracy_{win}$	0.2471	0.2539	0.1785	0.1821	0.1474	0.1042	0.0811	0.2000
$Accuracy_{place}$	0.5514	0.4932	0.4733	0.4517	0.3646	0.6117	0.2844	0.5433
Net gain	1.08	-13.98	-68.18	-85.63	-42.08	-4.81	-3.39	-3.30
Return/Bet	0.0774	-0.1554	-0.2384	-0.2710	-0.3691	-0.4004	-0.3769	-0.2753

Table 21 Testing Accuracy of model C + Odds across different classes

Class	*Class 1*	*Class 2*	*Class 3*	*Class 4*	*Class 5*	*Group 1*	*Group 2*	*Group 3*
$Accuracy_{win}$	0.2843	0.3949	0.2794	0.2251	0.2157	0.3142	0.3444	0.2250
$Accuracy_{place}$	0.7100	0.6811	0.5906	0.5714	0.4515	0.8425	0.5056	0.7692
Net gain	-3.81	-2.13	-50.34	-95.48	-29.90	0.16	-1.34	-5.21
Return/Bet	-0.2724	-0.0236	-0.1760	-0.3022	-0.2623	0.0132	-0.1492	-0.4340

From the data, model B has the highest training net gain when we consider the classes with positive net gain. Therefore, we can formulate a strategy to only bet on Class 1 and Group 3.

The performance of betting only on Class 1 and Group 3 is shown in the following table.

Class	*Class 1*	*Group 3*	*Overall*
Accuracy$_{win}$	0.2771	0.2825	0.2796
Accuracy$_{place}$	0.4979	0.6117	0.5504
Net gain	2.45	7.89	10.34
Return/Bet	0.1753	0.6573	0.3977

Table 22 Model B performance with the constructed strategy

When betting only on Class 1 and Group 3, model B can consistently make net gain with almost 40% return over bet. We claim that this model is an effective method to learn the different place probabilities of horses.

7.2 Discussion

In the results section it is observed that using odds data can increase prediction accuracy but have minimal impact on net gain and return over bet. This can be attributed to the fact that while adding odds data to input increases prediction accuracy, odds data may push model away from predicting horse with higher odds which means higher return. Therefore, although using odds data increases the prediction accuracy, the net gain and return over bet are not increased significantly.

In addition, it is observed that adding horse entity features in model C can further increase accuracy over model B but again does not result in higher net gain and return over bet. This is perhaps because when adding more horse entity features, the model gets closer to public intelligence and lead toward betting horses with lower odds, leading to a lower return.

Another possibility is that our model of 16 neurons per layer, while enough to learn from the input of model B (around 300), is too small to learn from the very high dimension (around 9000) input of model C. In this case, adding more neurons per layer may be able to improve performance.

8 Conclusion and Future Work

8.1 Conclusion

This report has detailed the process of using deep probabilistic programming predict horse racing. Though repeated experiments, we shown that horse racing prediction with deep probabilistic programming can beat public intelligence and generate net profit in circumstances much like those of LYU1703 [13].

8.2 Future Work

One of the main limitations is of our method is that we give equal importance to all training data. However, the utility/reward of predicting horse of high odds to win is much higher than predicting horses of low odds to win. Our current method fails the reflect the higher utility of predicting horses of high odds than predicting horses of lower odds. One way to overcome this limitation is to duplicate the entries according to odds. However, this increases memory consumption. Another approach is reinforcement learning with probabilistic inference [25], which we will investigate next semester.

Moreover, it is observed the in the experimental results that using more features has led to an decrease in net gain. We believe that this is due to insufficient neurons in the models per layer and using a wider model may help to alleviate this issue. In the next term, we would fine tune the different hyper parameters and attempt to improve our existing model.

Bibliography

[1] S. Russell and P. Norvig, Artificial Intelligence: a Modern Approach, Prentice–Hall, 1995.

[2] S. Thrun, W. Burgard and D. Fox, Probabilistic Robotics, MIT Press, 2005.

[3] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[4] K. P. Murphy, Machine Learning: A Probabilistic Perspective, MIT Press, 2012.

[5] J. M. Hernández-Lobato, M. A. Gelbart, R. P. Adams, M. W. Hoffman and Z. Ghahramani, "A General Framework for Constrained Bayesian Optimization using Information-based Search," *arXiv,* 2015.

[6] A. Shah and Z. Ghahramani, "Parallel Predictive Entropy Search for Batch Global Optimization of Expensive Objective Functions," *arXiv,* 2015.

[7] J. M. Hernández-Lobato, M. W. Hoffman and Z. Ghahramani, "Predictive Entropy Search for Efficient Global Optimization of Black-box Functions," *arXiv,* 2014.

[8] J. Burroni, "Extending Stan for Deep Probabilistic Programming," *arXiv,* 2018.

[9] D. Tran, "Deep probabilistic programming," *arXiv,* 2017.

[10] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," *arXiv,* 2014.

[11] A. Mnih and K. Gregor, "Neural Variational Inference and Learning in Belief Networks," *arXiv,* 2014.

[12] T. T. Cheng and M. H. Lau, "Predicting Horse Racing Result using TensorFlow," Department of Computer Science and Engineering, Hong Kong, 2017.

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

[14] The Editors of Encyclopaedia Britannica, "Encyclopædia Britannica," Encyclopædia Britannica, inc., 1 August 2018. [Online]. Available: https://www.britannica.com/sports/horse-racing. [Accessed 23 November 2018].

[15] Hong Kong Jockey Club, "Pari-Mutuel Pools," 1 August 2018. [Online]. Available: https://special.hkjc.com/racing/info/en/betting/guide_qualifications_pari.asp. [Accessed 23 November 2018].

[16] Hong Kong Racehorse Owner Association, "Breeding Strategies," 1 August 2018. [Online]. Available: http://www.hkroa.org/en/breeding_strategy.php. [Accessed 23 November 2018].

[17] Hong Kong Jockey Club, "Racing 101," 1 August 2017. [Online]. Available: https://entertainment.hkjc.com/entertainment/common/images/learn-racing/racing-academy/pdf/Racing_101_201708_Eng_Final.PDF. [Accessed 23 November 2018].

[18] M. Cox, "Balls and all: why Hong Kong horses are mostly geldings," South China Morning Post, 5 February 2017. [Online]. Available: https://www.scmp.com/sport/racing/article/2068269/balls-and-all-why-hong-kong-horses-are-mostly-geldings. [Accessed 23 November 2018].

[19] Hong Kong Jockey Club, "Racing 201," 1 August 2017. [Online]. Available: https://entertainment.hkjc.com/entertainment/common/images/learn-racing/racing-academy/pdf/Racing201_201708_Eng_Final.PDF. [Accessed 23 November 2018].

[20] Uber Pyro, "Pyro Examples and Tutorials," Uber Pyro, 23 May 2018. [Online]. Available: http://pyro.ai/examples/. [Accessed 23 November 2018].

[21] R. Ranganath, S. Gerrish and D. M. Blei, "Black Box Variational Inference," *arXiv,* 2014.

[22] D. Wingate and T. Weber, "Automated Variational Inference in Probabilistic Programming," *arXiv,* 2013.

[23] V. Mullachery, A. Khera and A. Husain, "Bayesian Neural Networks," *arXiv,* 2018.

[24] C. Blundell, J. Cornebise, K. Kavukcuoglu and D. Wierstra, "Weight Uncertainty in Neural Networks," *arXiv*, 2015.

[25] S. Levine, "Reinforcement Learning and Control as Probabilistic Inference: Tutorial and Review," *arXiv*, 2018.