AIDM7370 Group Project Report

# Horse Racing Prediction Using Neural Networks and Machine Learning

Group Name: Algorithm punks

Group members: FU Ziru (20466471)

GUO Qian (20453760)

Po Lai Kuen (20459742)

Chiu SingYuk (20431015)

Hong Kong Baptist University

# Abstract

Hong Kong has a long history of horse racing. Estimating horse racing results has been a very popular topic in the AI field. For that Hong Kong Jockey Club owns resourceful and comprehensive horse racing data, lots of experiments can be done. In this report, we present horse racing predictions using neural networks and some machine learning algorithms. This report can be explained in four main parts as follows. Firstly, we collect historical horse racing data and do exploratory data analysis and visualization. Secondly, we select useful features, build, train and test three different neural networks to predict horse racing results. The first ANN aims to predict the result class. The second ANN tries to classify the ranking position from the first place to the last place. The third one is to predict the time a horse finishes the race. Thirdly, we train and compare 3 machine learning algorithms for classification, which are random forest, KNN and LightGBM.

*keyword:* Horse racing prediction, Neural network, Machine learning, KNN

# 1.Introduction

Horse racing is Hong Kong’s most intriguing sport and entertainment due to this city’s gambling culture and the uncertainty of racing results. Hong Kong has over 150 years of horse racing history. Official racing began in 1884 with the establishment of Hong Kong Jockey Club, a non-profit organization that contributes its revenues to charity business and public projects (Who We Are -- The Hong Kong Jockey Club, n.d.). Every season, hundreds of races are held in Sha Tin or Happy Valley Racecourse. In each race, 8 to 14 horse runs in a row respectively for the championship and there are many betting strategies to predict the winner and win the bonus, such as “Win” (to predict the 1st horse in a race), “Place” (to predict 1st, 2nd and 3rd in a race), “Quinella” (to predict the 1st and 2nd in any order in a race) and so on.

Neural networks with a number of non-linear hidden layers are proved to be highly expressive to learn complicated relationships between their inputs and outputs (Srivastava, Nitish et al, 1989).

The exploration process in horse racing prediction is heading slowly. Alireza Khanteymoori employed many learning algorithms for horse racing prediction (Khanteymoori, 2010). Mehmet Akif Gulum used neural networks using graph-based features to predict racing results (Gulum, 2018, #).

The novelty of our work is that we firstly use three different neural networks to predict different horse racing results. Then, we compare three classification algorithms. We think the former papers did not explore the neural networks and machine learning algorithms as comprehensive as ours.

# 2.Methodology

## 2.1 Neural network

As shown in Figure 1, an artificial neural network (ANN) contains three layers basically, an input layer, a hidden layer and an output layer. The structure of an ANN is significant. The goal of ANN is to learn the hidden features of the dataset. Taking horse racing prediction as an example, if we have a dataset containing the horse condition, racecourse condition, etc and we know the corresponding horse won or lost. We want to sort the horses in a coming race into the “win” category and “lose” category. This is a classification problem. If we have the features and we know the time the corresponding horse finished the race. We want to predict the finishing time of each horse in a coming race. This is a regression problem.

图示

描述已自动生成

Figure 1

## 2.2 KNN

KNN’s classification idea is that if a sample has the k most similar samples in the feature space, the sample also belongs to this category.

## 2.3 LightGBM

Light Gradient Boosting Machine is based on Histogram's decision tree algorithm, using the deep leaf-wise decision tree growth strategy to improve accuracy.

## 2.4 Dummy classifer

Dummy Classifier is a classifier that makes predictions using simple rules. This classifier is useful as a simple baseline to compare with other (real) classifiers. Do not use it for real problems.

## 2.5 Logistic regression model

Logical regression is often used to estimate the probability that something belongs to a certain category.

## 2.6 Random forest

Random forest is an algorithm that integrates multiple trees through the idea of Ensemble Learning.

## 2.7 Confusion Matrix

A confusion matrix is a table that is often used to describe the performance of a classification model. The confusion matrix is used to evaluate KNN classifier, LightGBM and Random Forest. Figure 3 shows how precision, recall and f1-score are calculated.

表格

描述已自动生成

Figure 2

图示

中度可信度描述已自动生成

Figure 3

# 3.Experimental study

## 3.1 Dataset

### 3.1.1 Dataset description

In this project, three datasets were used. “races” and “runs” are downloaded from “Kaggle” (Graham Daley, 2019). They are historical racing information collected from 02 June 1997 to 05 June 2005.

“races” contains the information of each race.

‘runs’ describes the features of each horse and its racing result.

runs” adds a new column called “result\_class” results in “runs\_result\_class”. We separate the racing result to three groups, which are Place Group (1-4), Middle Group (5-9), and Slow Group (10-14).

“for\_visulization” is also from Kaggle (Lantana Camara, 2017). The dataset contains the race result of 1561 local races throughout Hong Kong racing seasons 2014-16, which is not used in ANN and ML part but only for visualization part.

‘actual\_17042021’ dataset contains the racing result on 17 April, 2021. We want to deploy our models on the latest racing and see the performance of our models.

### 3.1.2 Data preparation for the neural networks

Firstly, for the first ANN, we merge the “races.csv” and “run\_result\_class.csv”. For the second and third ANN, we merge the “races.csv” and “runs.csv”. We choose useful features. For the first ANN, we split our dataset into “X” dataset (dropping “result\_class” column) and “y” dataset (“result\_class”). For the second ANN, we split our dataset into “X” dataset (dropping “result\_1”... “result\_14” columns) and “y” dataset ( “result\_1”... “result\_14”). For the third ANN, we split our dataset into “X” dataset (dropping “finish\_time” columns) and “y” dataset (finish\_time”). Then, we do normalization. Then, we split our X and y into traning and test set.

## 3.2 Data analysis and visualization for some horse racing result factors

Relationship between winning rate & actual weight is showing in Figure 4. The ‘runs.csv’ and “run\_weight\_result” file are imported. It indicates that the winning rate is generally increasing when the actual weight is higher than 114 lbs, and has the highest winning rate when the actual weight is 133 lbs.

图表, 条形图

描述已自动生成

Figure 4. The relationship between winning rate & actual weight

Then, we analyze horse age with “heavyweight” and “lightweight groups. Figure 5 shows that the winning rate is decreasing with higher horse age in the “heavyweight” group.

图表, 折线图

描述已自动生成

Figure 5. The relationship between winning rate & horse age

As for the “lightweight” group, Figure 6 shows that the winning rate is increasing with higher horse age in this group.

图表, 折线图

描述已自动生成

Figure 6. The relationship between won rate & horse age

Then, we use the “for\_visualization.csv” file to find the best jockey and the best horse. As shown in Figure7, the best jockey is Moreira with the most number of wins and the highest winning rate. Figure 8 shows that Romantic Cash is the best horse.

图表, 散点图

描述已自动生成

Figure 7. Visualization of jockeys’ performance

图表

中度可信度描述已自动生成

Figure 8. Visualization of horses’ performance

Figure 9 says that low draws indeed have a considerable advantage. As the draw increases, the winning probability decreases.

图表, 饼图

描述已自动生成

Figure 9. Visualization of the Draw Bias Effect

In this part, we visualize the performance of jockeys and horses, and we find that the horse weight, horse age and draws indeed affect the result. In fact, there are many other factors which can influence the final result. The following part will illustrate how to predict horse racing results using ANNs and ML.

## 3.3 The first ANN

The structure of this ANN is displayed in Figure 10. This ANN has 12 neurons in the input layer, because we have 12 features to train and 2 hidden layers, each of which contains 64 neurons. The activation function we choose is the relu function. The output will be the result group of a certain horse.

By compiling the model, we choose “Adam” as an optimizer, “mse” as loss function and “mae” as “metrics”. Then we iterate the training dataset 30 times. As shown in Figure 11, the loss decreases gradually. The prediction results against ground-truth are shown in Figure 12. And we find that the prediction results match the actual result group.

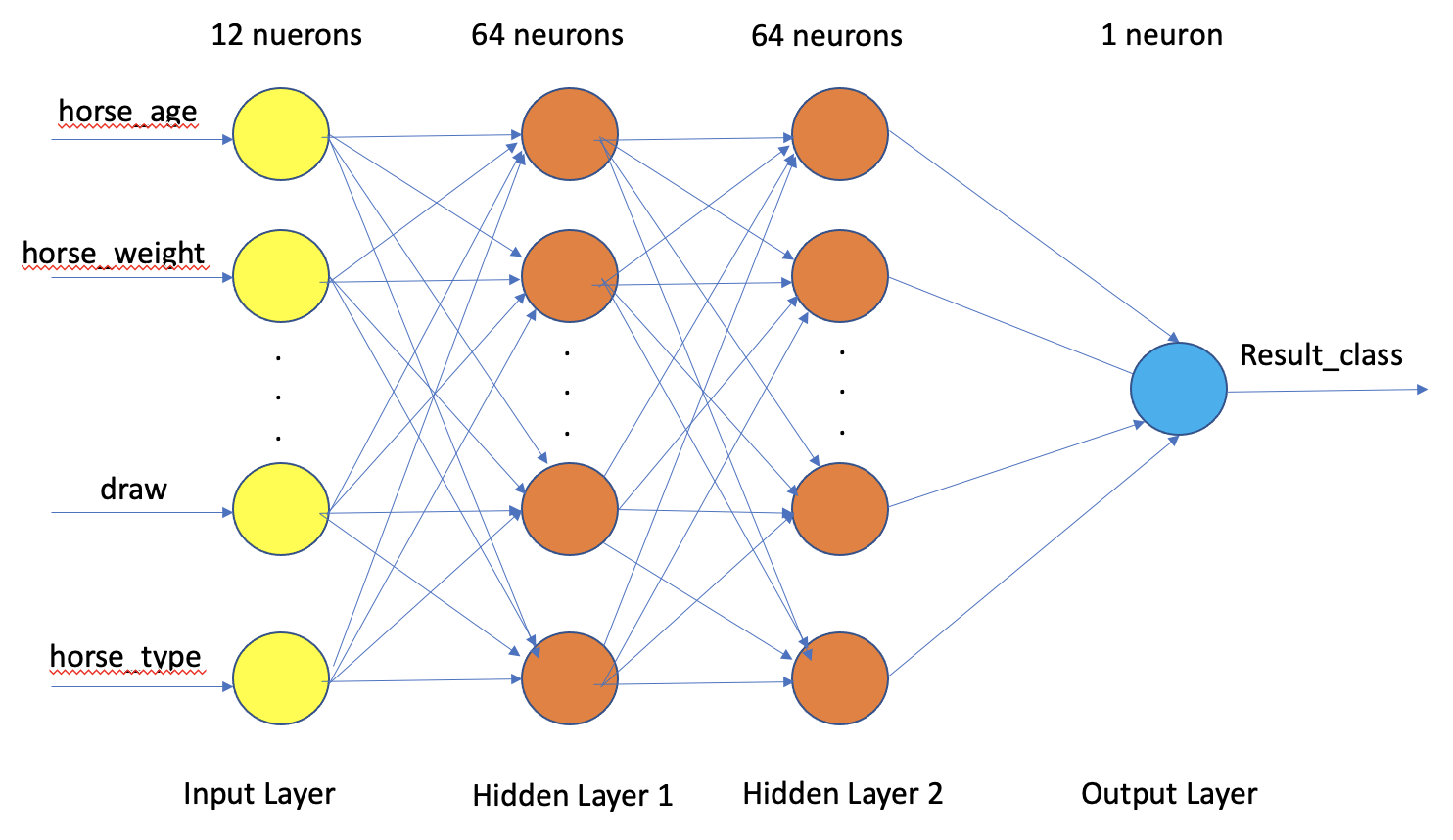


Figure 10

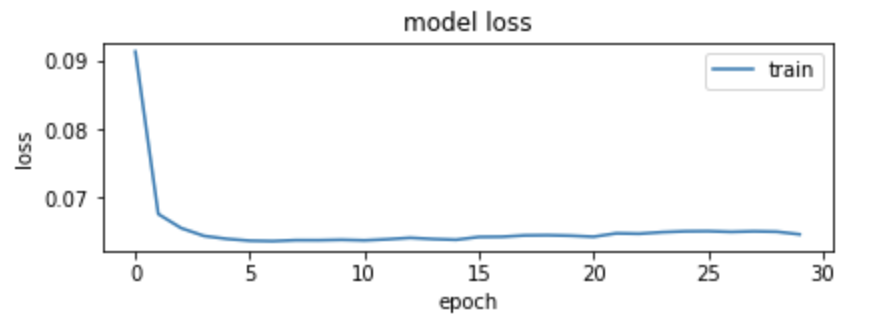


Figure 11

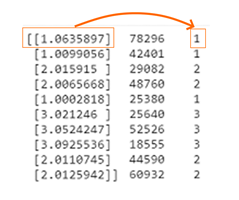


Figure 12

## 3.4 The second ANN

The structure of this ANN is displayed in Figure 13. The activation function we choose is sigmoid function. For that this is a classification problem, we put softmax function into the output layer. The output will be 14 categories from NO.1 to NO.14.

By compiling the model, we choose “Adam” as an optimizer, “Categorica lCrossentropy” as loss function and “accuracy” as “metrics”. Then we iterate the training dataset 30 times.

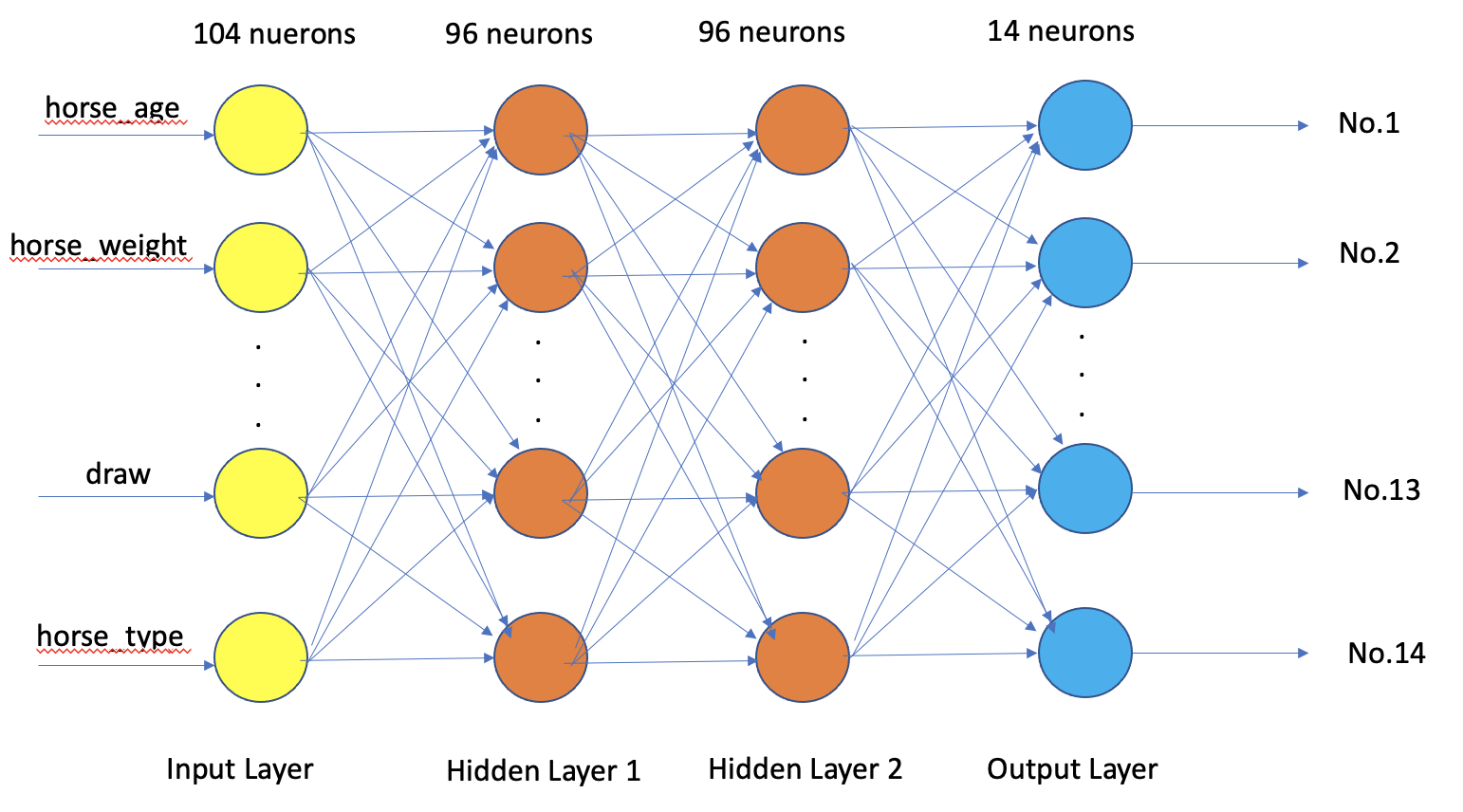


Figure 13

As shown in Figure 14, the model’s accuracy is approaching 0.3 and the loss decreases gradually. It is clear that predicting the exact ranking position is quite difficult.

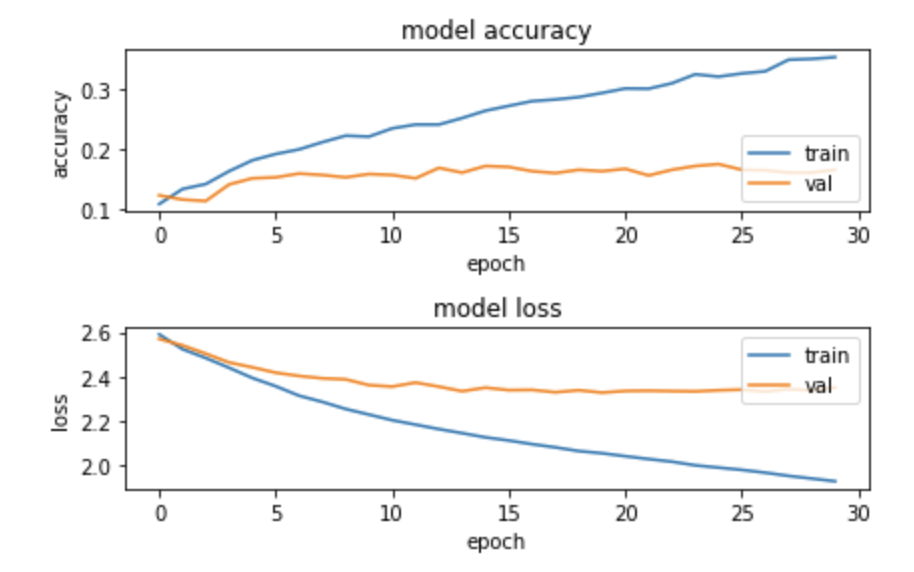
**

Figure14

## 3.5 The third ANN

The structure of this ANN is displayed in Figure 15. The activation function we choose is the relu function. The output will be the finishing time of a certain horse.

By compiling the model, we choose “Adam” as an optimizer, “mse” as loss function and “mae” as “metrics”. Then we iterate the training dataset 100 times.

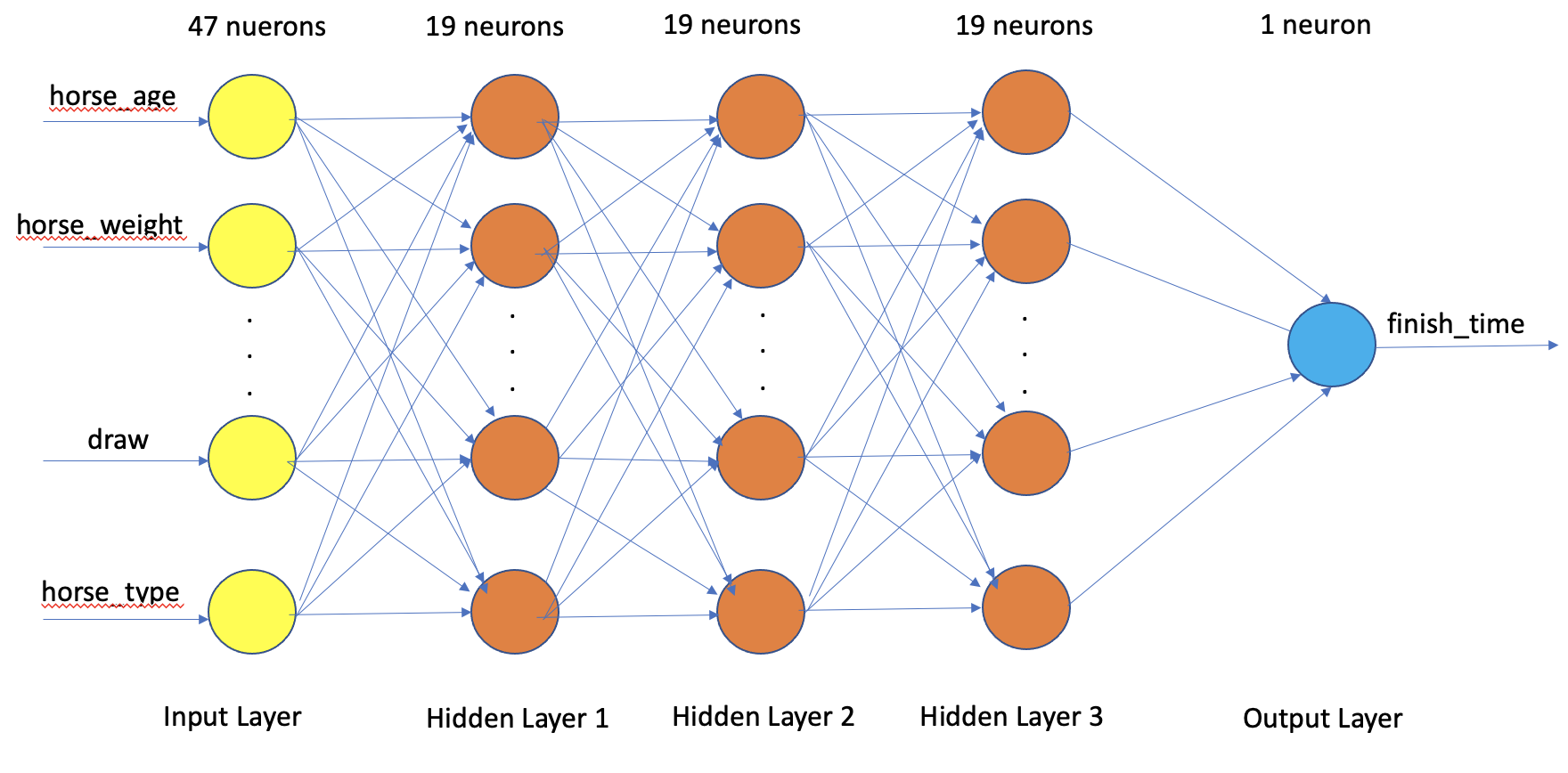


Figure 15

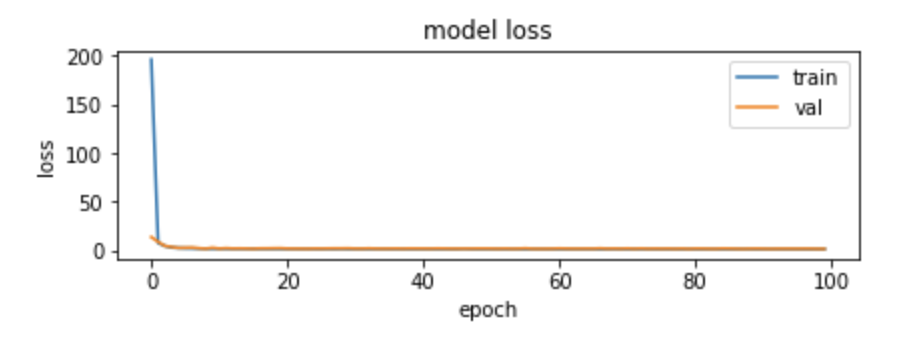


Figure 16

As shown in Figure 16, after testing the model, the model’s loss fluctuates around 1.5400011539459229, and the model’s accuracy is 0.7246764302253723. As shown in Figure 17, by deploying this model on the test dataset, the prediction is quite good.

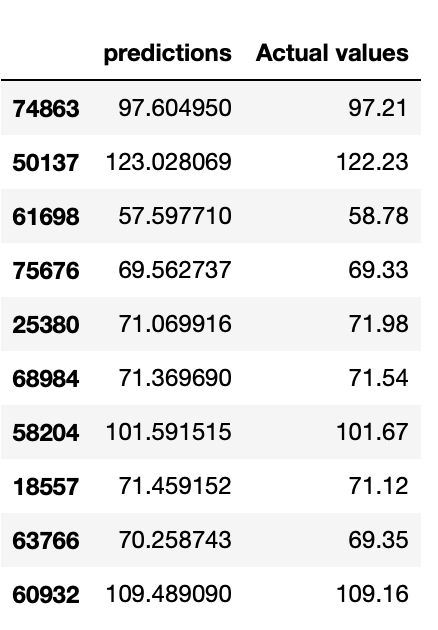


Figure 17

## 3.6 Train and test KNN classifier

Label ‘won’ as 1 and ‘lost’ as 0. The distribution of the labels is shown below as Figure 18. Then, split dataset as X (dropping “won” column) and y (containing only “won” column).

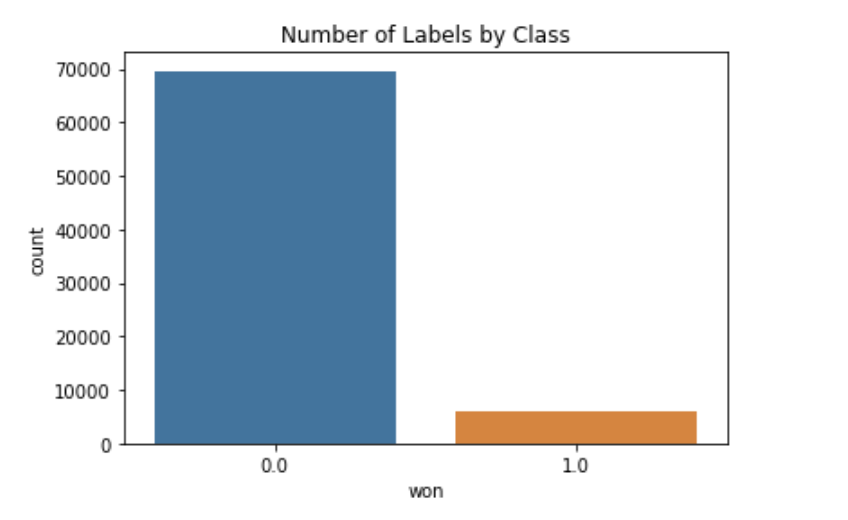


Figure 18

As the outcome shown in Figure 19, when using original data for sampling, the accuracy of the KNN classifier is 0.92 which is the highest. At this time, the precision of 1.0 is 0.32. Explain that the KNN classifier using original data can sort out winners more accurately.

表格

描述已自动生成

Figure 19. KNN classifier classification report for original data

*表格

描述已自动生成*

Figure 20. KNN classifier classification report for under-sampling data

表格

描述已自动生成

Figure 21. KNN classifier classification report for over-sampling data

## 3.7 Train and test LightGBM classifier

We use “LightGBM” from “sklearn” package of Python. As the outcome shown in Figure 23, LightGBM classifier shows a higher precision with under-sampling data, the model accuracy is about 0.88. At this time, the precision of 1.0 is 0.30. Explain that the Light GBM classifier using under-sampling data can sort out winners more accurately.

表格

描述已自动生成

Figure 22. LightGBM classifier classification report for original data

*表格

描述已自动生成*

Figure 23. LightGBM classifier classification report for under-sampling data

*表格

描述已自动生成*

Figure 24. LightGBM classifier classification report for over-sampling data

## 3.8 Train and test dummy classifier

We use the DummyClassifier to always predict “not win” as a baseline model because we know that in the “won” column there are much more 0 (not win) than 1 (win).

表格

描述已自动生成

Figure25. Dummy classifier classification report

From the Figure 25, we got an accuracy score of 92% without training a model. Accuracy is the ratio of correct predictions.

## 3.9 Train and test random forest classifier

We use “RandomForest Classifier” from “sklearn” package of Python. As the outcome shown in Figure 26, the accuracy of the Random Forest Classifier is 0.92. At this time, the precision of 1.0 is 0.39.

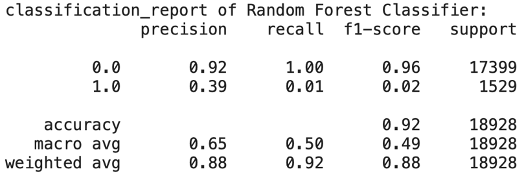


Figure 26

## 3.10 Forest classifier classification report

## 3.10 Train and test Logistic Regression

## We use logistic regression method from “sklearn” package of Python. The outcome of Logistic Regression is shown in Figure 27.

表格

描述已自动生成

Figure 27. Logistic Regression classifier classification report

Our accuracy score is the same as compared to the dummy classifier above. This tells us that accuracy might not be our best option for measuring performance. In such an unbalanced problem, accuracy is meaningless: A very dumb model predicting always zeros would have great accuracy, to the detriment of the predictive power of class 1, which has precision and recall equal to zero.

# 4.Conclusion

In this report, we represent three different neural networks to predict the result classes, ranking positions and finishing time respectively. We find that it is easier to predict the result classes and the finishing time, and it is very difficult to predict the exact ranking positions. In this report, we represent three different neural networks to predict the result classes, ranking positions and finishing time of a certain horse respectively. We have tuned many parameters and selected the most suitable functions. We know that different numbers of layers and neurons and selection of functions could affect the final result. We also find that it is easier to predict the result classes and the finishing time, and it is very difficult to predict the exact ranking positions from NO.1 to NO.14.

As for the machine learning algorithms, Random Forest Classifier with original data, LightGBM with under-sampling data and KNN classifier with original data reaches the highest precision to sort out the winning horses.

# 5.Discussion

Betting on horse racing needs to take many factors into consideration. In this report, when training our ANNs, we only focus on the horses’ conditions and the racecourses’ conditions. However, a jockey’s performance is also a key factor. In the future, we could probably take jockeys’ conditions into account, in order to train a more sophisticated ANN. Moreover, we only select 3 classification machine learning algorithms to compare. If we use other methods like Bayesian Classifier and Support Vector Machine, we might get better results. Last but not least, for future research, we want to compare the similarities among races and try to apply pattern matching methods to find the most confident races for betting.

# Reference

Graham Daley. (2019). *Horse Racing in HK*. Kaggle. https://www.kaggle.com/gdaley/hkracing

Gulum, M. A. (2018, 05). Horse racing prediction using graph-based features. *Electronic Theses and Dissertations*. https://doi.org/10.18297/etd/2953

Khanteymoori, A. (2010, 06). *Horse racing prediction using artiﬁcial neural networks*. researchgate. https://www.researchgate.net/publication/228847950

Lantana Camara. (2017). *Hong Kong Horse Racing Results 2014-17 Seasons*. Kaggle. https://www.kaggle.com/lantanacamara/hong-kong-horse-racing

Srivastava, Nitish et al. (1989). *Dropout: A Simple Way to Prevent Neural Networks from Overﬁtting*. Neural Networks. https://www.cs.cmu.edu/~epxing/Class/10715/reading/Kornick\_et\_al.pdf.

*Who we are -- The Hong Kong Jockey Club*. (n.d.). The Hong Kong Jockey Club. https://corporate.hkjc.com/corporate/english/who-we-are/index.aspx