THE CHINESE UNIVERSITY OF HONG KONG

FINAL YEAR PROJECT REPORT

Predicting Horse Racing Result with Machine Learning

Author:
Yide LIU, Zuoyang WANG

Supervisor:
Prof. Michael LYU

A final report submitted in fulfillment of the requirements for the final year project

in the

LYU 1703
Faculty of Engineering
Department of Computer Science and Engineering

November 30, 2017

Declaration of Authorship

We, Yide LIU, Zuoyang WANG, declare that this report titled, “Predicting Horse Racing Result with Machine Learning” and the work presented in it are our own.

The report is a collaborated work of both authors. In this report, Yide is responsible for Chapter Introduction, Data Preparation and Conclusion. He also provides model performance metrics and figures . Zuoyang is responsible for Chapter model design and results, contributing to investigating data and figures. Both authors share their views and understandings in the discussion section and revise the report for cross proving.

"No wife can endure a gambling husband; unless he is a steady winner."

Thomas Dewar

THE CHINESE UNIVERSITY OF HONG KONG

Abstract

Faculty of Engineering

Department of Computer Science and Engineering

BSc degree in Computer Science

Predicting Horse Racing Result with Machine Learning

by Yide LIU, Zuoyang WANG

Neural networks with a large number of parameters are very powerful machine learning systems. While neural networks has already been applied to many sophisticated real-world problems, its power in predicting horse racing results has yet not fully explored. Horse racing prediction is closely related to betting and the netgain is considered a benchmark of performance. This project offers an empirical exploration on the use of neural networks in horse racing prediction. We constructed augmented racing record dataset and raceday weather dataset and examine architectures in a wide scale. We showed that neural networks can identify the relationship between horses and weather and our models can achieve state-of-the-art or competitive results. Comparisons are provided against traditional models such as win-odds betting and performance-based betting and also learning models in LYU1603.

Acknowledgements

We would like to express my special thanks of gratitude to our supervisor Prof. Michael Lyu as well as our advisor Mr. Edward Yau who gave us the golden opportunity to do this wonderful project on the topic 'Predicting Horse Racing Result with Machine Learning', which also helped us in doing a lot of Research and we came to know about so many new things and we are really thankful to them.

Secondly, we would also like to thank our friends who helped us a lot in finalizing this project within the limited time frame.

Contents

List of Figures

List of Tables

Chapter 1

Overview

This topic on final year project is predicting horse racing result with machine learning, throughout this report we will demonstrate the work done during the first semester. This chapter offers a brief overview to this final year project and introduction to the topic. Moreover, it provides related work and previous approaches on the horse racing predictions. In the end, it introduces the difficulties in predicting horse racing results.

1.1 Introduction

Neural networks with number of non-linear hidden layers is proved to be highly expressive to learn complicated relationship between their inputs and outputs (Srivastava et al., 1989). Pragmatically, neural networks are shown to present its learning power in machine learning and becoming the dominant approach for many problems (Srivastava et al., 2014). Although they are introduced first in late 1950s (Widrow and Hoff, 1959), the increasing functionality of computer hardware and the use of graphics processing unit (GPU) enables the training processing recently. In the latest research in visual object recognition and then following in other research fields, neural networks are designed to goes deeper in layers(He et al., 2015) and a term called "Deep Learning", i.e. training a deep neural network with multitudinous layers,is appearing often in both the academic and the public society.

However, when researching on a new field of studies, traditional approach begins studying networks expressive power from very few layers (Huang et al., 2016). In visual object recognition, for example, begins with a primitive network called LeNet (LeCun et al., 1998) consisted of 5 layers and recent study of Highway Network (Srivastava, Greff, and Schmidhuber, 2015) and Residual Networks (He et al., 2015) surpass 100 layers. Latest research in neural networks in this year shows that very deep networks with exceeding 1000 layers have been studied and employed (He et al., 2016). While it helps to go deeper in network structure, the study of neural networks requires researcher to start from a very beginning of smaller version. These approaches are accord with the nature of neural networks: the exceeding number of neurons and parameters, the large carnality of hyper-parameter space, the appropriate score function and insidious structure issues. Since training a deep neural network takes days and months to run (Vanhoucke, Senior, and Mao, 2010), it is reasonable to train network with simple structure in order to accelerate the research progress.

Conventionally, neural networks are developed in steps to target sets of well-known academic problems. Neural networks are fully explored in those classical problems: text classification (Kim, 2014; Zhang, Zhao, and LeCun, 2015), and sentimental analysis (Santos and Gatti, 2014; Ouyang et al., 2015) in natural language processing, pattern recognition and object detection (Ren et al., 2015; Szegedy, Toshev, and Erhan, 2016) in computer vision, auto-encoding (Lange and Riedmiller, 2010) and noisy-encoding (Graves, Mohamed, and Hinton, 2013) in information theory. In spite of the promising performance in classical problems, the power neural network in other real-world problems is still under exploration. The complexity and unclear relationship makes it difficult to express the relationships.

One of these undetermined real-world problems is horse racing. The prediction of horse racing result has been a popular research topics in recent years. However, the research in this fields make little progress over these years. Few paper is published in academic domain after the prediction problem is firstly introduced in 2008. Two similar studies reviewed the power of neural network using different optimization techniques and performed finishing time predictions base on previous horses racing records (Snyder, 1978b; Williams and Li, 2008). Last year, LYU1603 (Tung and Hei, 2016) worked with two different approaches: binary classification on winning and logistics regression on horse finishing time. Their models realized positive net gains only with a threshold over 95% on betting confidence. Those studies provides different approaches to interpret horse racing problems but in contrast reveals a lack of understanding in horse racing predictions.

The horse racing events, while they are commonly considered a special kind of game, follows similar characteristics shared with stock market predictions where futures performances are related to precious and current performances to some extent. On the other hand, unlike games of perfect information such as GO[1] and PENTAGO[2] (Méhat and Cazenave, 2011), the optimal value function, which determines the outcome of a game, is not well-defined (Silver et al., 2016). While the horse racing prediction problem being a mixture of imperfect information and stochastic randomness (Snyder, 1978a), previous naive approaches fails to capture the critical information and produce few promising results. To the best of our knowledge, current horse racing prediction is limited and the results are under satisfaction.

In this final year project, we scrutinize features of horse racing events and predict horse racing results directly through finishing time. The rest of this report is organized as follows: Chapter2 illustrates how first-hand data is collected and structured. Moreover it provides prudent statistical analysis on related features and data standardization. The model design and configurations along with comparison models are presented in Chapter3. In Chapter4, we review the prediction metrics and present the experimental results. Our understandings and interpretations on the results are discussed in Chapter5. In the end, we conclude the accomplishment achieved in this term and offer possible research directions in next semester.

1.2 Background

Horse racing is a sports to run horses at speed. Horse racing is not only a professional sports but also of a beloved entertainment of betting in Hong Kong. Every season, hundreds of races are held respectively in Shatin and Happy Valley racecourses at different tracks and distance. In each race, 8-14 horses runs in a row for the fastest and various bet types are created for entertainment on the result of the races.

Horse racing events are managed by the Hong Kong Jockey Club (HKJC). HKJC is a non-profit organization to formulate and develop horse racing, sporting and betting entertainment in Hong Kong. Moreover, it is the largest taxpayer and community benefactor in Hong Kong. It holds a government-granted monopoly in providing pari-mutuel betting on horse racing. In the history of horse racing in Hong Kong, the HKJC plays a essential role in promotion and regulation and combines the betting entertainment into this sports. "With strict rule enforcement, betting fairness

[1] https://en.wikipedia.org/wiki/Go_(game)
[2] https://en.wikipedia.org/wiki/Pentago

and transparency, the HKJC has taken the Hong Kong racing to a world-class standard and also earned itself an enviable global reputation as a leading horse racing organization."

1.2.1 Pari-mutuel betting

Betting is the most fascinating attraction of horse racing by the nature of pari-mutuel betting system. Pari-mutuel betting is a betting system in which the stake of a particular bet type is placed together in a pool, and the returns are calculated based on the pool among all winning bets (Riess, 1991).
Dividend is divided by the number of winning combinations of a particular pool. Winners shares the percentage of pool payout proportional to their betting stakes and taxes are reducted from the dividend in a particular ratio.

1.2.2 Types of bets

There are multiple types of bets of a single race as well as multiple races. The following tables from the HKJC website provides an explanation of each betting type.

Single-race Pool

Single-race Pool	Dividend Qualification	
Win	1^{st} in a race	Demo Video
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	Demo Video
Quinella	1^{st} and 2^{nd} in any order in a race	Demo Video
Quinella Place	Any two of the first three placed horses in any order in a race	Demo Video
3 Pick 1 (Composite Win) **Winning Trainer (Composite Win)** **Winning Region (Composite Win)**	Composite containing the 1^{st} horse in a race	Theme Site
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	Theme Site
Quartet	1^{st}, 2^{nd} , 3^{rd} and 4^{th} in correct order in a race	Theme Site

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

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
T-T Auto Pick	**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

FIGURE 1.1: Types of bets

1.2.3 Methodology

Intriguingly, there are many possible ways to interpret horse racing results and a few are studied in previous studies. In this research, we takes finishing time approach to model horse performance. Moreover, we try to bet on the best horse with the fastest estimated finishing time.

It is worth mentioning that it is still an open topic to model horse racing results and different approaches cannot avoid the deficiency in predictions and betting. In this section, we provides pros and cons for most common approaches in other researches.

Finishing time

One way to deal with this problem is to build a regression model. In this project, we train a supervised neural network regressor on the finishing time and then ranks each horse base on the calculated predicted time. The model takes individual records of race information into account and then learn the relationship in a general way. However, due to the nature of this approach, the predicted time of horses in a race is less reasonably distributed. In some cases, max-min finishing time reaches up to 10 seconds.

Horse to win the race

Another way to solve the problem is naturally predicting a horse to win or not. However, directly doing binary classification of win or lose is unfeasible because the dataset will be unevenly distributed with less than 10% horses marked with "win" (or equally "1").

A tricky approach is to directly investigate the logistics of the results and rank them in every race. Equivalently, given the credibility of a model, we can design a betting strategy to ensure positive expectation net gain.

Horse ranks

The third way to predict the horse racing result is directly predicting the horse ranks. However, due to the same issue mentioned above, ranks in a races can be duplicated and hence it is unreasonable to view the problems in this way.

1.2.4 Objective

In this project, we restrict our discussion on "win" and "place" bets, which is the most effective way to reflect the model efficiency and explore the relationship between betting gain and prediction accuracy. Our objective is to create a model to predict horse racing results and beat public intelligence in betting net gain.

The following table revisits the "win" and "place" bets and defines the prediction accuracy of each type ($Accuracy_{win}$, $Accuracy_{place}$).

TABLE 1.1: WIN and PLACE bet revisit

Bet Type	**Dividend Qualification**	**Accuracy Definition**
win	1^{st} in a race	Correct win bets out of all bets
place	1^{st}, 2^{nd}, 3^{rd} in a race	Correct place bets out of all bets

To distinguish the prediction accuracy of finishing time and $Accuracy_{win}$, $Accuracy_{place}$, we first define the *avg_loss* of which represent the average mean-square-error (MSE) between the prediction finishing time and the ground truth. We later define the *Accuracy* to be the reciprocal of *avg_loss*.

In the following chapters, we shows that the *Accuracy* actually is **not correlated** to any of $Accuracy_{win}$, $Accuracy_{place}$ and it is one of the main finding and difficulty in this project.

Chapter 2

Data Preparation

Recent research in various areas has shown that neural networks works well on different fields attributed to its flexibility of in learning complex patterns. However, this flexibility may lead to serious over-fitting issue and performance degradation in small datasets in the case of memerization effects (Arpit et al., 2017) (e.g. more parameters than training examples) when a model "memorizes" all training examples.

In general, Neural network is prone to work better on large-scale and complete datasets especially when the model works on raw features (like our project). Yet, dataset constructed by LYU1603 only involves limited features and out-of-date. Therefore, instead of adapting our models to take in data by LYU1603, we built 2 large datasets, racing records and raceday weather, for our experiments. Our datasets involves all records originating from 2011 and automatically collects the newest race information.

This chapter first illustrates the approach to collecting data and then describes the datasets and corresponding database structures. After that, it provides careful analysis on significant features taken in our model. The last section shows the preprocessing steps applied in this research.

2.1 Data Collection

Historical data are provided online by several commercial companies[1] and HKJC official website. A possible way to constructed our database is to purchase from such companies. However, due to the financial conditions and the quality of these

[1] hkhorsedb

sources, we switched to obtain the data by ourselves from the web.

Base on the characteristics on two datasets, we design two handy crawler accordingly from: HKJC offical and Timeanddate. Historical data from 2011 to up-to-date data are collected and maintained in MySQL server. Our crawling system can automatically collect the latest racing and weather result and make it possible to for our model to predict future races.

2.2 Datasets

2.2.1 Horse Racing Record

The horse racing record dataset contains all racing data from 2011. Each row in the dataset represents a record keeping information of a certain horse in a selected race. The dataset contains 63459 records from 5029 races taken place in Hong Kong. The following table describes the useful features that directly crawled from HKJC website.

TABLE 2.1: Useful racing features collected from HKJC website

Feature	Description	Types	Values
raceyear	Race year	Record Index	-
racemonth	Race month	Record Index	-
raceday	Race day	Record Index	-
raceid	Unique id of a race	Record Index	-
location	Location where a race take place	Categorical	ST, HV
class	Class of horses meaning strength of a horse	Categorical	1 to 5
distance	Distance	Categorical	1000, 1200, 1400, 1600, 1650, 1800, 2000, 2200
course	Track	Categorical	*Over 8 distinct values
going	Track condition	Categorical	* Over 10 distinct values
raceno	Race number in a race day, abstraction of race time	Categorical	1 to 8
horseno	Number assigned by HKJC to a horse	Categorical	1 to 4-14
horseid[a]	Unique code of a horse	Categorical	*Over 1000 distinct values
jockeycode[a]	Unique code of a jockey	Categorical	*Over 50 distinct values
trainercode[a]	Unique code of a trainer	Categorical	*Over 30 distinct values
draw	Draw of a horse	Categorical	1 to 4-14
actualweight	Weight of gears carried by a horse	Real Value	-
horseweight	Weight of a horse	Real Value	-
winodds	"WIN" Odds of a horse	Real Value	1-00
place_odd	"PLACE" Odds of a horse	Real Value	1-99
recordid	Unique id of a record	Record Index	-
place	Place of a horse **LABEL**	Categorical	1 to 4-14
finishtime	Finishing time of a horse **LABEL**	Real Value	-

Features generated after a race is abandoned since they cannot be collected before a race
Complete Categorical values are listed in Appendix

Apart from features directly crawled from HKJC website, we extracted advantageous features imitating professional tips. Base on the initial measured data, we derived 2 informative features, old place and weight difference, that explore the trend between two consecutive races of a horse. Moreover, we explicitly created a feature called "dn" to indicate whether the race is taken place during daytime or nighttime. The following table provides descriptions for these features.

In comparisons with LYU1603, the extracted data is brainstormed and added by us originally.

TABLE 2.2: Extracted racing features

Feature	**Description**	**Types**	**Values**
dn	Day or Night	Categorical	D, N
old_place	Place of a horse in the last race	Categorical	1 to 4-14
weightdiff	Weight difference of a horse since the last race	Real Value	-

2.2.2 Weather

We obtained weather dataset from Timeanddate. The dataset is obtained from historical data recorded in two observatory located near Shatin and Happy Valley racecourses, containing 5029 weather information on every race in Horse Racing Record database. Each row of weather dataset indexed by the racetime. The following table illustrate information contained in the weather datasets.

The weather data is first taken into model inputs in this research and statistical analysis is conducted in following section.

TABLE 2.3: Weather features

Feature	Description	Types	Values
raceyear	Race year	Record Index	-
racemonth	Race month	Record Index	-
raceday	Race day	Record Index	-
raceno	Race number in a race day	Record Index	1 to 8
location	Location where a race take place	Record Index	ST, HV
temperature	Temperature when a race start	Real Value	-
weather	Weather condition when a race start	Categorical	*Over 28 distinct values
wind_speed	Wind seed when a race start	Real Value	-
wind_direction	Wind direction	Categorical	*16 combination of diretions
humidity	Humidity when a race start	Real Value	-
moon	Moon phase of race day	Real Value	0-28[a]

[a] cycle of moon phase is 28 Complete Categorical values are listed in Appendix

2.3 Data Analysis

2.3.1 Horse Racing Features

In this section, we will carefully examine the relationship between features in horse racing dataset and horse performance.

Class

Class is a strutured feature manually created by HKJC by horse rating. There are 5 classes in addition to Group and Griffin races. In handicap races, the standard upper rating limit for each class will be as follows:

TABLE 2.4: Class and rating standard in handicap races

Race Class	Standard Upper Rating Limit
1	120
2	100
3	80
4	60
5	40

The following figure from HKJC website illustrates the complete class system of HKJC races. It is worth mentioning that our project only focus on handicapped races among class 1 to 5.

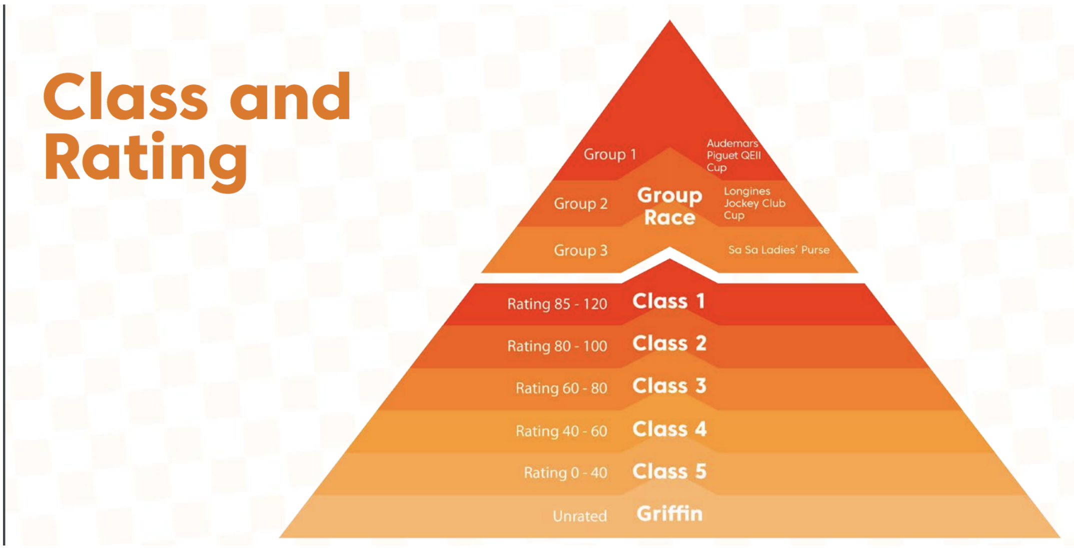


FIGURE 2.1: Complete class system

Horse rating is calculated by HKJC base on recent performance of a horse, however, due to the limitation of historical data the rating of a horse in race time is not recorded. One approach is to intimidate a horse power standard to train our models. Yet last year project has shown that ELO system only has little contribution to training performance. In result, we only use class to do anticipation.

The following table illustrates the correlations between class and horse performance. When finishing time is normalized by distance, we can observe a trend that the higher class of a horse, the quicker it finishes.

TABLE 2.5: Class Correlation Matrix

	finishtime	place	class
finishtime	1.000000	0.480295	0.328883
place	0.480295	1.000000	0.017722
class	0.328883	0.017722	1.000000

*Finishtime is normalized by distance to represent horse performances.

Win odds

Win odds reflects the public intelligence on the expectation of horses in a single races. In Pari-mutuel betting system, the larger the win pool of a horse, the lower of win odds of a horse. The correlation between both finishtime and place shows that public intelligence is wise enough to predict horses.

Although following public intelligence improves the model predictions, the betting system by design reversely makes betting difficult: betting on the lowest rate will result in negative net gain in statistical expectation. Therefore, enhancing model prediction is necessary and challenging.

TABLE 2.6: Winodds Correlation Matrix

	finishtime	place	win odds
finishtime	1.000000	0.480295	0.227527
place	0.480295	1.000000	0.472042
win odds	0.227527	0.472042	1.000000

*Finishtime is normalized by distance to represent horse performances.

Weight

Most races are handicaps and more weights are added to the stronger runners in order to equalize the chance of winning. The allocated weight is determined by the horse recent performance (proportional to the horse rating assigned by HKJC).

In our research, we inspect relationships between performance and weight-related features. There are three features in total: carried weight, horse weight and actual weight. Carried weight is the weight of gear attached to a horse in handicap race.

Horse weight is weight of the horse itself. Actual weight is a extracted feature which is the sum of carried weight and horse weight. Actual weight represents the real weight of a horse.

The following correlation matrix portrays linearity between all weight features.

TABLE 2.7: Weight Correlation Matrix

	finishtime	**place**	**carried weight**	**horse weight**	**actual weight**
finishtime	1.000000	0.480295	0.005194	-0.057819	-0.056780
place	0.480295	1.000000	-0.099052	-0.030117	-0.039850
carried weight	0.005194	-0.099052	1.000000	0.037630	0.138260
horse weight	-0.057819	-0.030117	0.037630	1.000000	0.994897
actual weight	-0.056780	-0.039850	0.138260	0.994897	1.000000

*Finishtime is normalized by distance to represent horse performances.

The table illustrates that neither of weight features is closely related to finishing time or place. The statistics is quite convincing since the handicapped rules are designed to adjust the horse power. Yet we take a deeper look into these features trying to understand whether such features will help in predictions. We randomly select two horses and analyze the correlation between weight features and their performances.

TABLE 2.8: Weight Correlation Matrix of "A003"

	finishtime	**place**	**carried weight**	**horse weight**	**actual weight**
finishtime	1.000000	0.629141	0.702231	-0.725612	-0.574200
place	0.629141	1.000000	0.190517	-0.555372	-0.595973
carried weight	0.702231	0.190517	1.000000	-0.641935	-0.337157
horse weight	-0.725612	-0.555372	-0.641935	1.000000	0.938297
actual weight	-0.574200	-0.595973	-0.337157	0.938297	1.000000

*Finishtime is normalized by distance to represent horse performances.

TABLE 2.9: Weight Correlation Matrix of "L169"

	finishtime	**place**	**carried weight**	**horse weight**	**actual weight**
finishtime	1.000000	0.244387	0.160243	-0.027932	0.031567
place	0.244387	1.000000	0.038202	0.149508	0.147911
carried weight	0.160243	0.038202	1.000000	0.105043	0.448122
horse weight	-0.027932	0.149508	0.105043	1.000000	0.936099
actual weight	0.031567	0.147911	0.448122	0.936099	1.000000

*Finishtime is normalized by distance to represent horse performances.

We can conclude from the above table that weight features are working differently with horse performance of individual horses. The horse "A003" may suffer from obesity in this case and losing weight help it perform in a large scale. On the contrary, the horse "L169" may be of well health and weight features have minor influence on its performance.
Another possible explanation for the above matrices is that the horse "L169" is in a less competitive class while the horse "A003" competes with strong opponents and the change in weights influence "A003's" performance greatly.

Weight difference

Weight difference is the difference of horse weight. The feature to some extend reflects the health conditions of a horse and in this part we try to identify the relationship.

TABLE 2.10: Weight difference Correlation Matrix

	finishtime	**place**	**weight difference**
finishtime	1.000000	0.480295	0.061861
place	0.480295	1.000000	0.073577
weight difference	0.061861	0.073577	1.000000

*Finishtime is normalized by distance to represent horse performances.

Combined with the analysis in the preceding sections, we believe that individual weight difference influences the performance in different ways even though it shows no relation with the overall performance.

Old place

Old place can be strictly defined as the place of a horse in its previous game. Consistent performance relies heavily on the strength of the horse nature. Even though horse racing is affected by a number of factors (for example, running with better opponents), a horse with an excellent place in the previous race tends to runs consistently as shown in the following correlation matrix. We claim that the nature of horse is invariant to minor environment changes in most cases.

TABLE 2.11: Old place Correlation Matrix

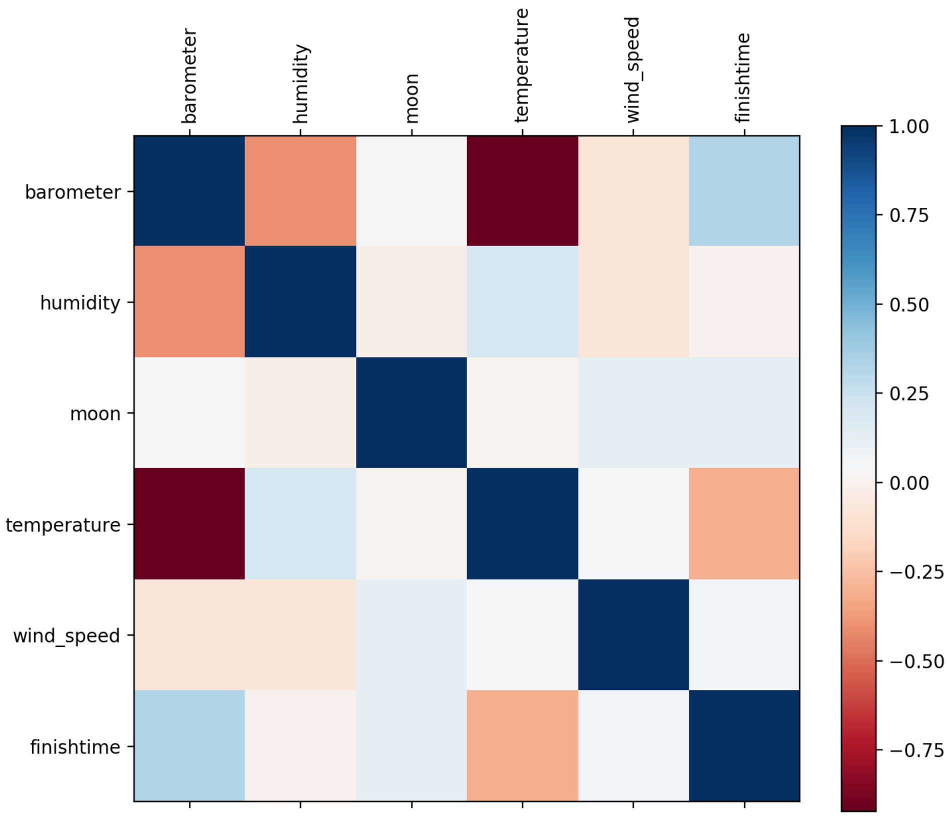
	finishtime	**place**	**old place**
finishtime	1.000000	0.480295	0.144424
place	0.480295	1.000000	0.236155
old place	0.144424	0.236155	1.000000

*Finishtime is normalized by distance to represent horse performances.

2.3.2 Weather Features

Weather conditions attributes to horse racing finishing time to some extent. In our research, taking weather into account is shown to help enhance predicting performance.

When it comes to overall performance of the horses, average finishing time varies in different weather conditions. In common sense, horses tends to be longer in raining days than in sunny days. Moreover, horses are prone to run faster under warmer temperature. Empirical results on average finishing time against different weather conditions show that common sense proves out to have high credibility.



*Finishtime is normalized by distance to represent horse performances.

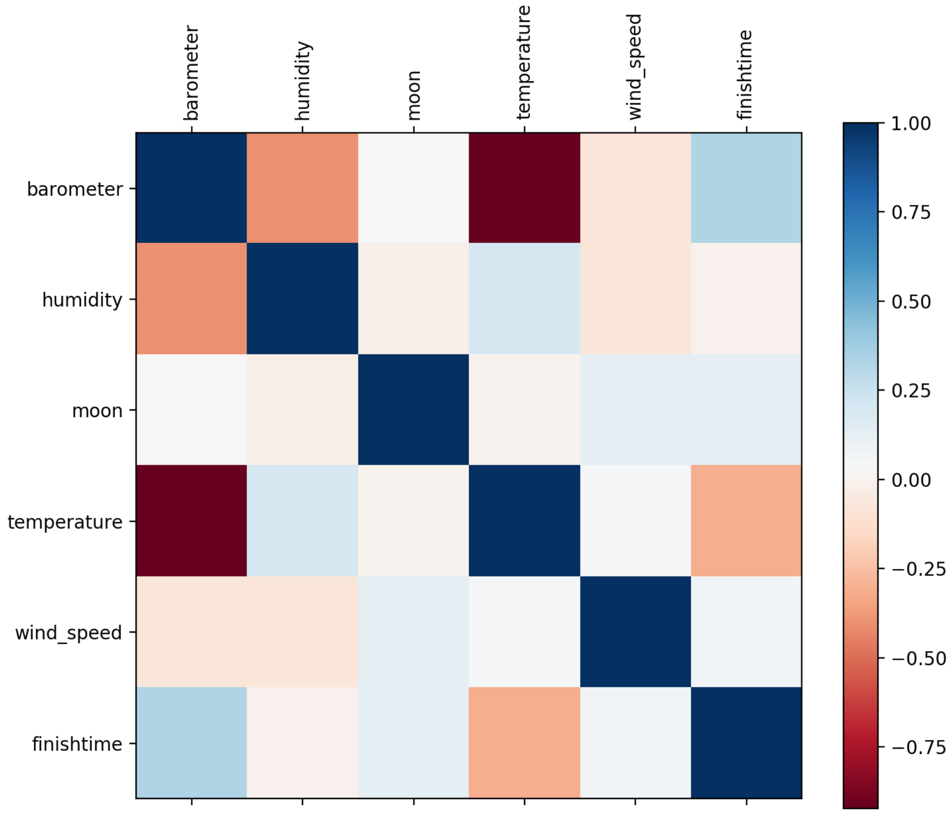
FIGURE 2.2: Average finishing time against different weather conditions

One possible explanation for the fluctuation in finish time is that, a change in weather regardless of its form, may have subtle influence on horse nature.

On the one hand, the weather can indirectly determine horses performances by slightly changing the course environment. For example, the condition of a race track plays an important role in the performance of horses in a race. A slight flutuation in humidity and raining can make a great impact in track surface density, porosity, compaction and moisture. In a result, the horse tends to run in a different speed.

On the other hand, the rise in humidity and temperature can affect the health condition and further emotions of horses. Horses are astute on their surroundings. When the environment changes quickly, they can easily get agitated and flee away. In general belief, horses themselves are responsible for the minor change of environment and the results differs naturally under different weather.

The following figure shows the correlations between the horses finishing and weather conditions collected in our research. The finishing time is shown to have strong correlations with some of weather features, such as temperature and humidity. However, relationship between other features is wanting to be discovered. Moreover, individual performances with different weather is hard to show due to the large quantity and it remains for our model to discover.



*Finishtime is normalized by distance to represent horse performances.

FIGURE 2.3: Average finishing time correlation with weather

2.4 Data Preprocessing

In accord with LYU1603, we split the datasets into two training and test sets. The training set contains race records from 2011 to the end of 2014 and the test set contains records between 2015 and 2016.

2.4.1 Real Value Data

Normalization is a common requirement of for machine learning tasks in order to make training less sensitive to the scale of features(Sola and Sevilla, 1997). The most commonly used is $z - score$ normalization that scales individual samples to have unit norm, i.e. given the real value data matrix X, where the rows represent the individual records and the columns represent the features, the normalization transform the matrix into $X_{normalized}$, such that

$$X_{normalized,j} = \frac{X_j - mean(X_j)}{std(X_j)}$$

where X_j is the j^{th} column of matrix X.

In this research, we perform *z-score* normalization on real value data columns in our datasets. The mean and standard deviation is calculated base on training set and then apply to test set to avoid information leak.

2.4.2 Categorical Data

Categorical data are challenging to train and always mask valuable information in a dataset. It is crucial to represent the data correctly in order to locate most useful features in the dataset and downgrade the model performance. Multiple encoding scheme is available across the market. One of the most simple one is through one-hot encoding in which data is encoded into a group of bits among which the legal combinations of values are only those with a single high (1) bit and all the others low (0) (Harris and Harris, 2010). However, one-hot encoding suffers from its high cardinally and the feature space can grow exponentially making it unfeasible to train.

Thanks to *feature_column* APIs in Tensorflow, advanced encoding schemes is provided to map those categorical data into hash slot automatically.

Two useful APIs[2][3]are provided to encode those data into sparse matrix in reasonable space. *categorical_column_with_hash_bucket* allows users to distributes sparse features in string or integer format into a finite number of buckets by hashing; *categorical_column_with_vocabulary_list* allows users to map inputs of similar format to

[2]https://www.tensorflow.org/api_docs/python/tf/feature_column/categorical_column_with_vocabulary_list
[3]https://www.tensorflow.org/api_docs/python/tf/feature_column/categorical_column_with_hash_bucket

in-memory vocabulary mapping list from each value to an integer ID. Subsequently, a multi-hot representation of categorical data is created by *indicator_column* or embedded manually with assigned buckets by *embedding_column*.

In our research, we research in different combinations of *feature_columns* in terms of performance and running time. Due to the nature of horse prediction, complex networks with high feature dimensions that requires large amount of training time are unpractical since the future race data is given one day before the race day. To balance out the performance and training time, we choose *categorical_column_with_vocabulary_list* and *indicator_column* to represent categorical models in our network models.

Chapter 3

Model Architecture

In this chapter, we introduce our models in this final year project. The model is designed to solve the regression in predicting finishing time of a horse using the neural networks with 2 non-linear hidden layers. Optimization is performed by back-propagation along gradients to minimize the mean square error(MSE) of predictions. (REFERENCE FROM ZHANG)

3.1 Deep Neural Network Regressor

The key design of our model is an adaptation from traditional network classification model. Instead of using an activation layer (typically logistic function or softmax function) in classifier problems, our model treats hidden layers output as final output and use the identity function as activation function. Therefore, it uses the mean-square-error as the loss function, and the output is a set of continuous values.

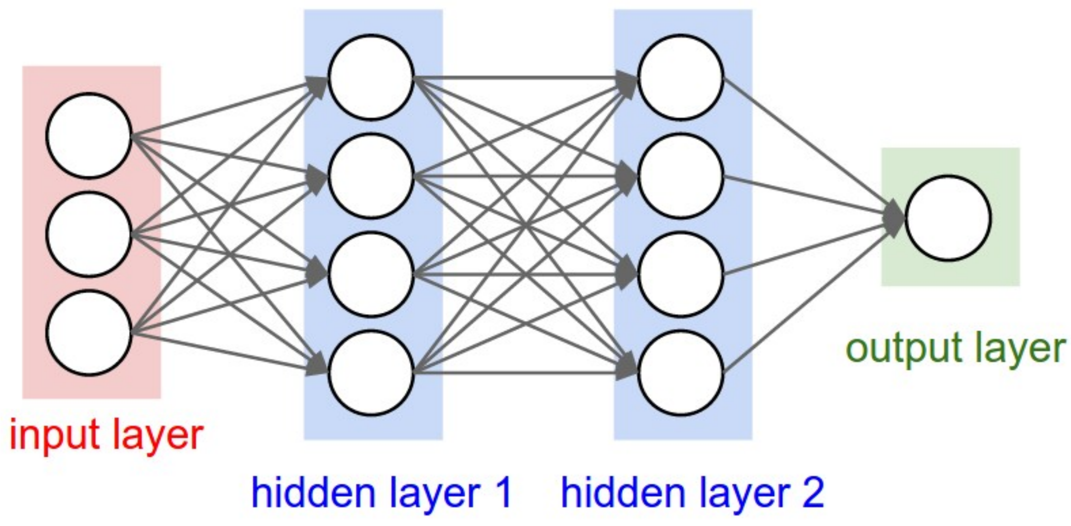


FIGURE 3.1: Deep Neural Network Regressor

3.1.1 Methodology

"Any class of statistical models can be termed a neural network if they use adaptive weights and can approximate non-linear functions of their inputs. Thus neural network regression is suited to problems where a more traditional regression model cannot fit a solution."

3.1.2 Configurations

For a DNN model, the first thing to decide is the structure, including the number of layers and the batch size. In term of number of layers, we use 2 layers which is commonly employed in DNNs.

The batch size of a model is the number of flow units in the model. We arbitrarily choose to use the popular setting 128*128 in the end because theoretically it can achieve a balance between performance and computational efficiency.

Then we need to decide the data frame for training and testing our models. In order to be consistent and comparable with the previous teams, i.e. LYU1603 and LYU1604, we split the data, use data from 2011 to 2014 to train and data from 2015 to 2016 to test the models.

Also the amount of training steps needs to be decided. A few experiments were conducted on training steps of 10k, 100k and 1m to find out the relatively best steps, which turned out to be 10k. It shows that the training model may be very easy to overfit, thus more steps (>10k) would lead to a worse result.

Number of Steps	Noodds_noweather	noodds_weather	odds_noweather	odds_weather
10k	4.025	3.603	4.347	3.263
100k	4.291	4.697	4.819	3.668
1m	5.192	5.221	5.088	4.281

TABLE 3.1: Experiments on the number of training steps

As Table 3.1 shows, the models that trained 10k steps have an advantage over the 100k and 1m ones, so 10k is accepted to be part of the standard configuration to conduct further experiments.

3.1.3 Evaluation Standard

Before comparing the performance of the models, a well-grounded evaluation standard is essential. The first criteria is the loss retrieved from the model itself. We apply the default loss evaluation function provided by Tensorflow here.

The second criteria is the accuracy of predictions. Since the models themselves only predict the finish time of each horse, we group the predictions by races and obtain the winning horse of each race. Then the actual accuracy of the predictions can be drawn.

The third criteria is the overall net gain after simulating the real bets over 2015-16. Since it is a real world question, there is never a better way than evaluate a model by putting all its predictions into the real scene.

Chapter 4

Experimental Results and Discussion

In this chapter, the result of all the experiments will be shown and interpreted from more than one dimension. Also, through combining the models to each other, we are searching for the best betting strategy that claims for the most net gain. A conclusion will be drawn on basis of all the data.

4.1 About the Experiments

The purpose of the experiments is to figure out which factors can really improve the prediction. Since the DNN models are like black boxes that cannot be seen clearly from outside, the experiments are essential to help understand the question.

The first factor of the experiments is the division of the data sets. There are 2 racecourses in Hong Kong now. One in Sha Tin and one in Happy Valley. Races taking place in the two location are believed to be different. We wonder whether the "divide-and-conquer" strategy can be applied here to help solve this question, so we both train a model of the whole data set and train separated models of the subsets grouped by different locations.

The second factor is the odds. The winning odds of each race can be retrieved and fed to the models, and by intuition is closely related to the winning horse. However team LYU1603 finally decided not to use this feature. To make this clear, both models with and without the "winodds" feature will be trained and compared in the experiments.

The third factor is the weather information. Historical weather information of the racecourses are grabbed from web in advance, including features like temperature, wind speed and humidity. It is not clear that to what extent these data would help improve the prediction, so models will be train with and without these data separately.

To sum up, 3 binary factors are presented, thus 8 models are trained correspondingly in the experiments.

4.2 Experimental Results

4.2.1 Single Model

Notation

In order to keep the table neat and readable, a special notation is used here, which uses three binary digits to represent the model. For example, "Model 000" means the model is NOT divided by location, NOT including "winodds" nor "weather" in the feature, while "Model 110" means the model is divided by location, including "winodds" but excluding "weather". Also, for those models starting with "1", since each of them they involves 2 sub-models, the first value refers to the data of Sha Tin and the second refers to those of Happy Valley.

A General Look at the Results

Models	Model 000	Model 001	Model 010	Model011	Model100	Model 101	Model 110	Model 111
Loss	515.2	461.2	556.4	417.7	583/575	527 / 536	629 / 577	652 / 589
$Accuracy_{win}$	0.08367	0.07029	0.08090	0.10742	0.08355	0.07560	0.08488	0.07560
$Accuracy_{place}$	0.08753	0.10031	0.09063	0.09461	0.09284	0.09461	0.09372	0.09991
Net gain	-1087	-991	-1378	-568	37/-1005	-1088/-1579	655/ -917	339/-1724

TABLE 4.1: Model performance data

Model 011 has the best loss when predicting the finishing time of each horse. For most of the models, including weather as a feather leads to a significant decrease in loss. However, including win odds does not improve the result; in contrast, 3 of the 4 comparisons show a increase in loss. In terms of the division of data set, basically the divided models perform worse than the corresponding undivided models.

While concerning $Accuracy_{win}$, Model 011 wins again. However, after comparing accordingly, most models show the pattern that weather does not help improve the accuracy this time. Similarly, it seems that both win odds and the division of data sets do not make an obvious difference as well, since most of the models remains essentially unchanged.

If we bet completely accordingly to the predictions' suggestions, we will get the net gain shown above. Unfortunately, all of the models lose money over the whole year. However, Model 110 loses the least money among the eight. Meanwhile, the models from different groups show different patterns. To the models using the undivided data set, the weather data has a positive impact on the prediction, while to the divided ones it is mostly negative. Also, if the weather data is excluded, dividing the data sets is significantly better than the opposite. Moreover, the bets on the races in Sha Tin gain much more money than those on the races in Happy Valley.

There are a few possible reasons for some of the above phenomenon which are anti-intuitive. First, after dividing the data set into 2 subsets, the data in each set may not be abundant enough for the model with lower loss to be trained out. Second, though theoretically adding feature should be helpful to improve prediction accuracy, it does not work out as expected, possibly because this involves a lot of randomness. Last, due to some mysterious unknown reasons, betting on the races in Sha Tin is better than on those in Happy Valley, which can be a useful hint for further studies.

Detailed Betting Trends of the Models

In this part we will show the detailed betting logs through the whole year of each model.

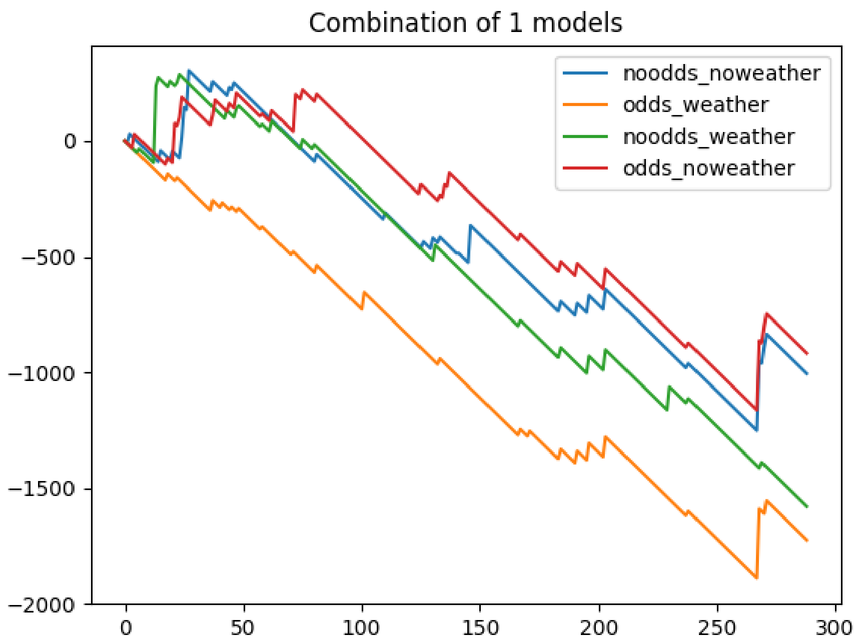


FIGURE 4.1: Net gain of divided data set (HV)

This figure shows the net gain changes of different models of the races in HV.

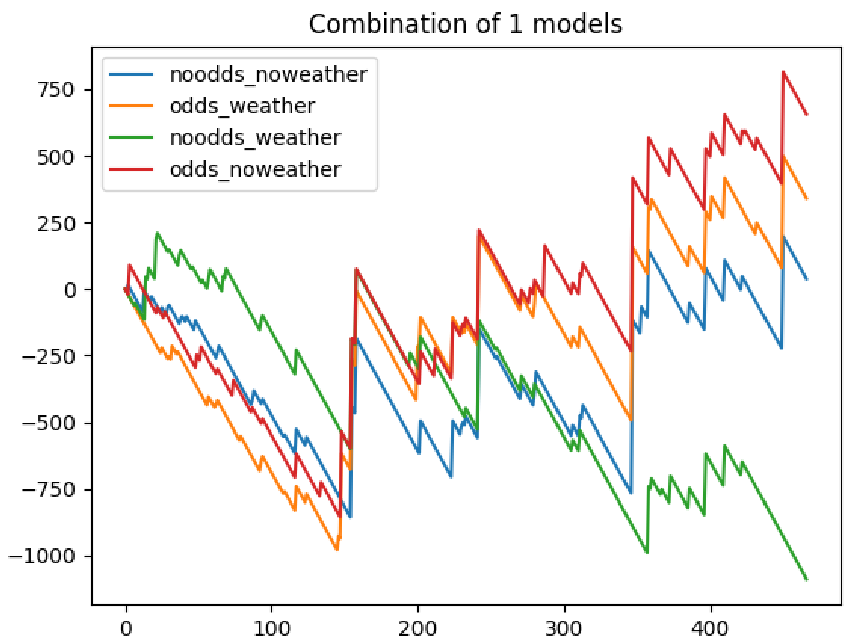


FIGURE 4.2: Net gain of divided data set (ST)

This figure shows the net gain changes of different models of the races in ST.

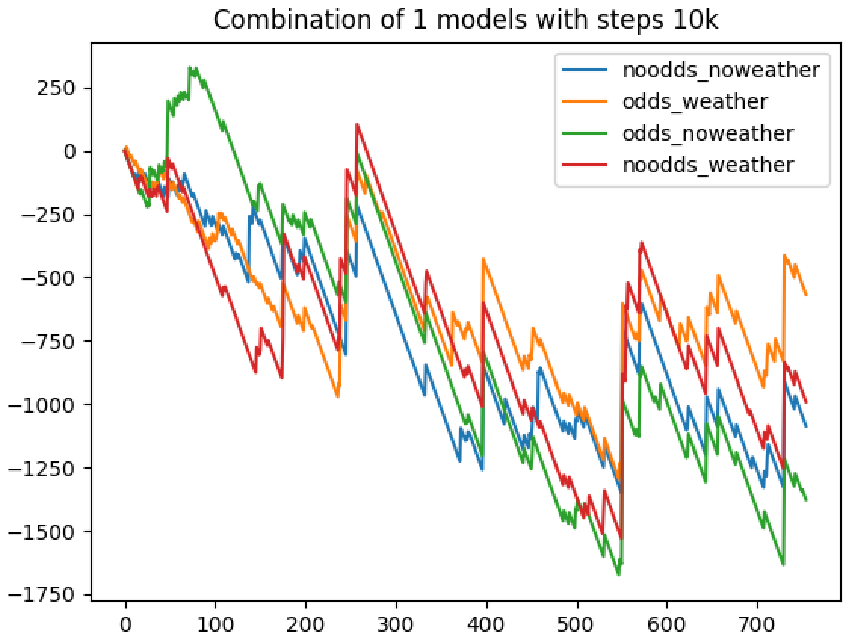


FIGURE 4.3: Net gain of undivided data set

This figure shows the net gain changes of different models of all the races.

Summary

The results of each models are shown and some primitive conclusions can be drawn. First, the connections among loss, accuracy and net gain are weak, which means pursuing a lower loss or a higher accuracy, just as people usually do in machine learning and data mining, does not really work on this topic. Instead of that, net gain need to be calculated accordingly to justify whether a model is good or not. Also, predicting horse racing results is a real-life question that involves much randomness. As a result, some of the ways that should theoretically improve the models do not work as well as expected. Moreover, betting without any consideration or filtering is stupid, and leads to an unsatisfactory outcome, so we are going to try some other ways to improve it.

4.2.2 Combinations of models

To make full use of the models and to generate more reliable predictions, a set of combinations of the models is tested. The basic assumption is that, if more than one model conform to each other, this piece of prediction is more reliable and worthy of betting. The following shows the results of the experiments.

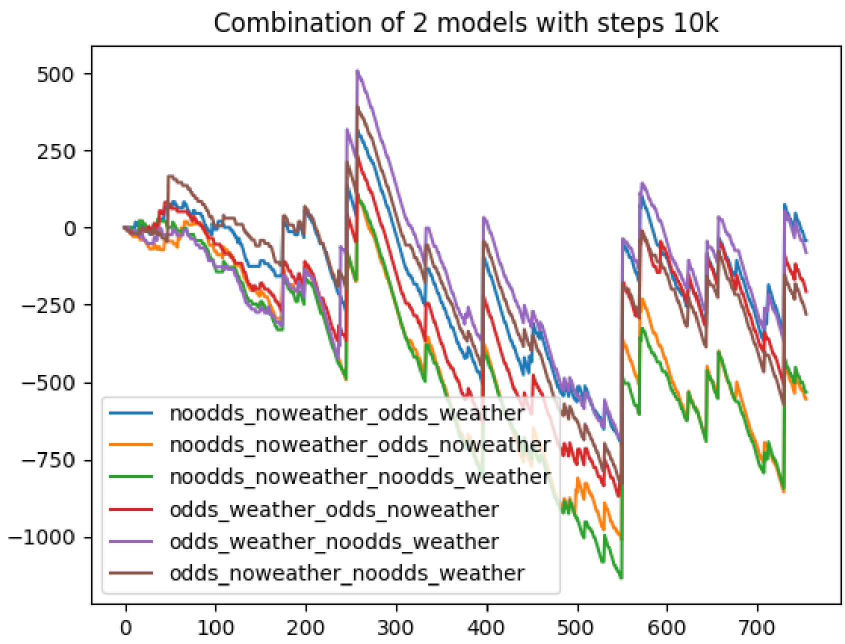


FIGURE 4.4: Net gain of combination of 2 models from undivided data set

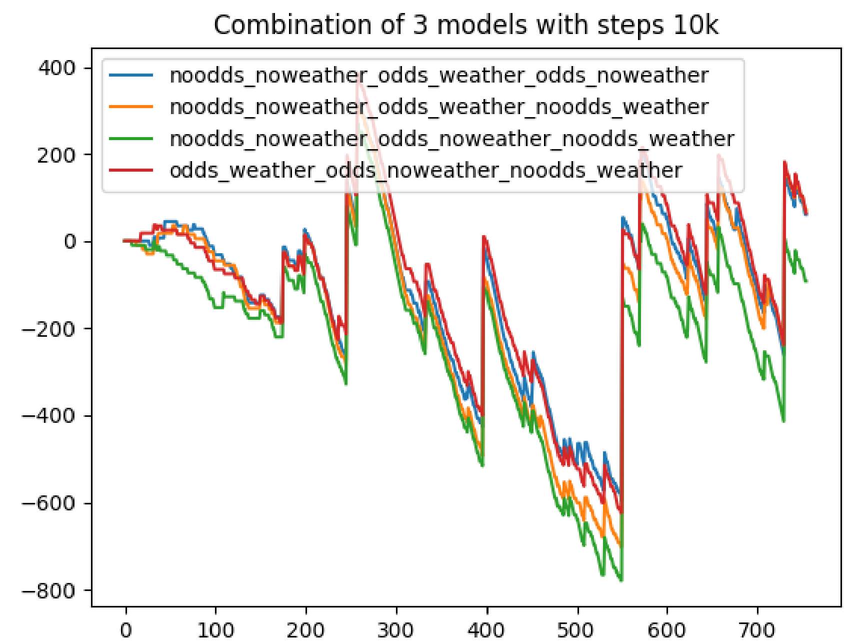


FIGURE 4.5: Net gain of combination of 3 models from undivided data set

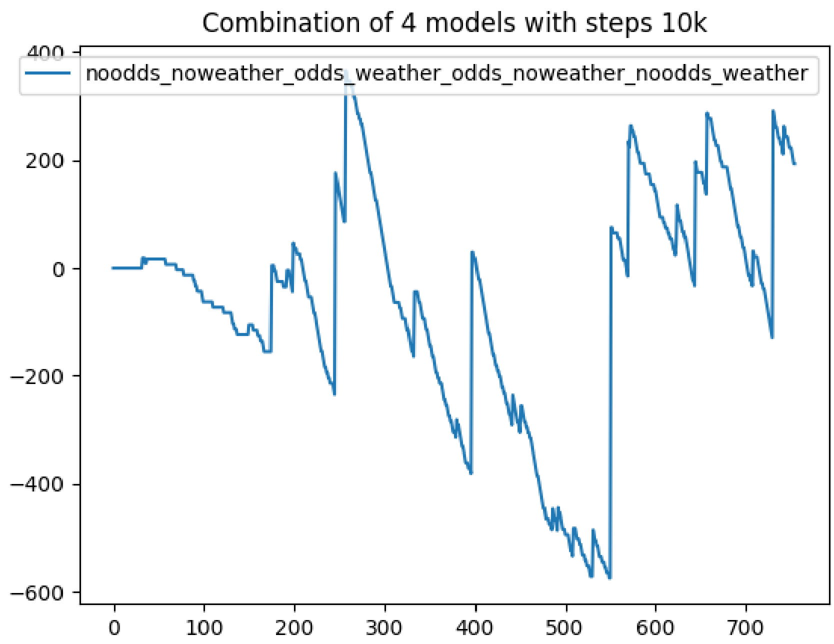


FIGURE 4.6: Net gain of combination of 4 models from undivided data set

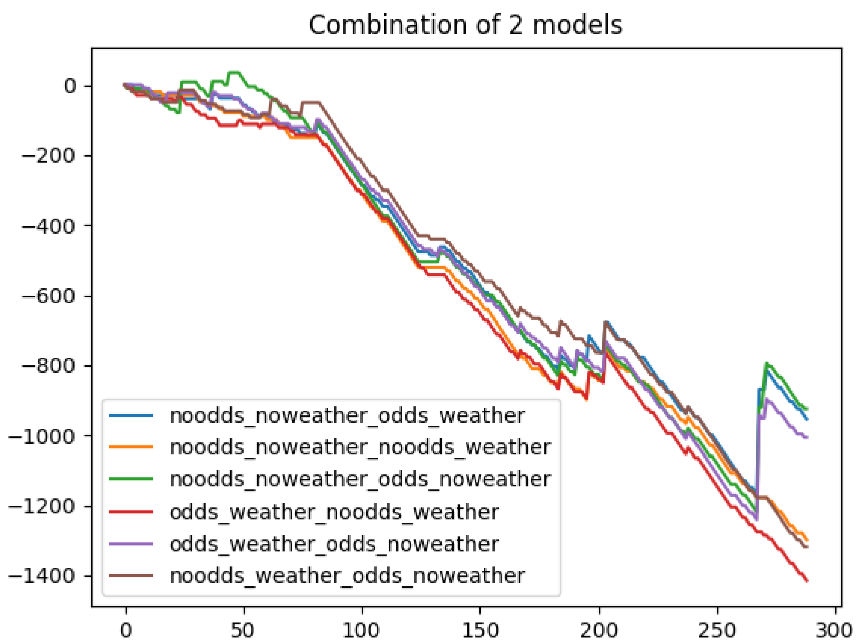


FIGURE 4.7: Net gain of combination of 2 models from divided data set (HV)

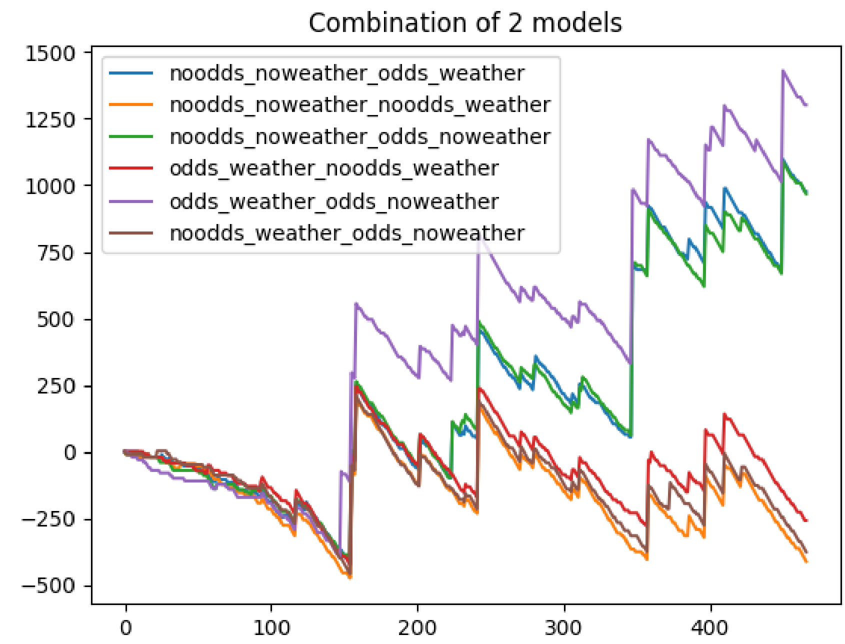


FIGURE 4.8: Net gain of combination of 2 models from divided data set (ST)

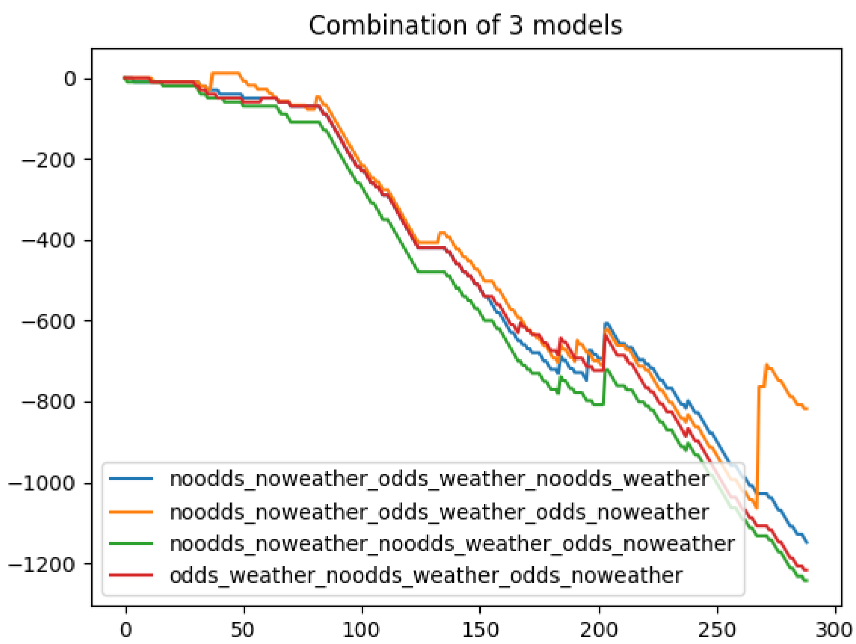


FIGURE 4.9: Net gain of combination of 3 models from divided data set (HV)

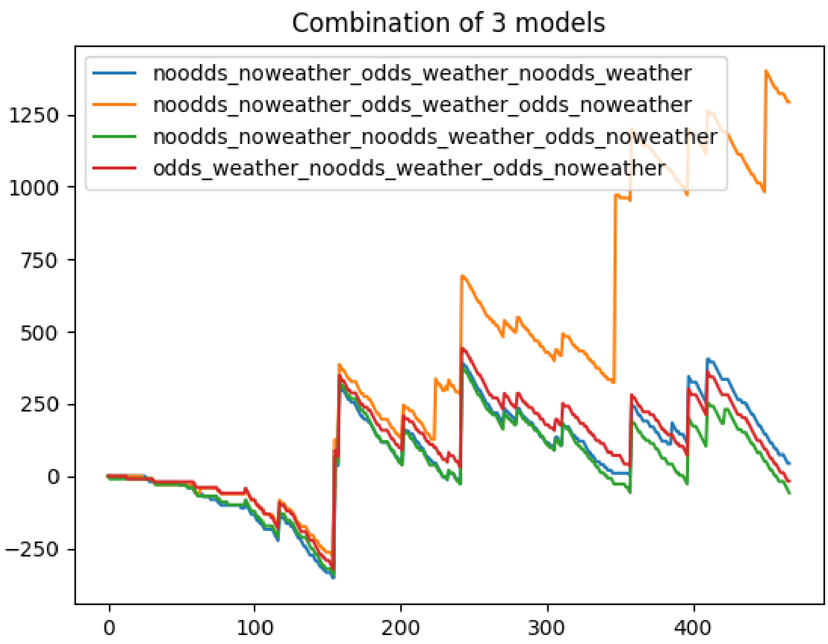


FIGURE 4.10: Net gain of combination of 3 models from divided data set (ST)

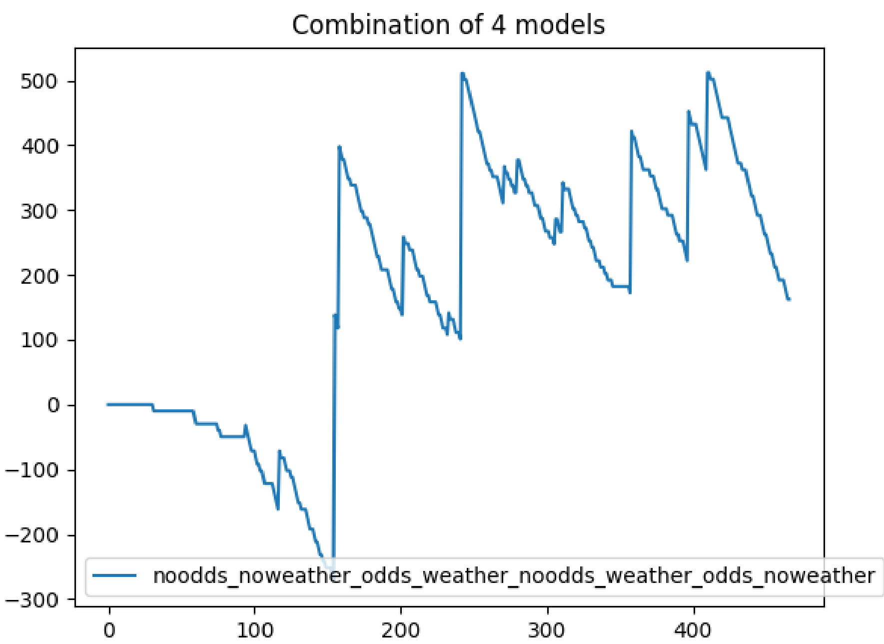


FIGURE 4.11: Net gain of combination of 4 models from divided data set (HV)

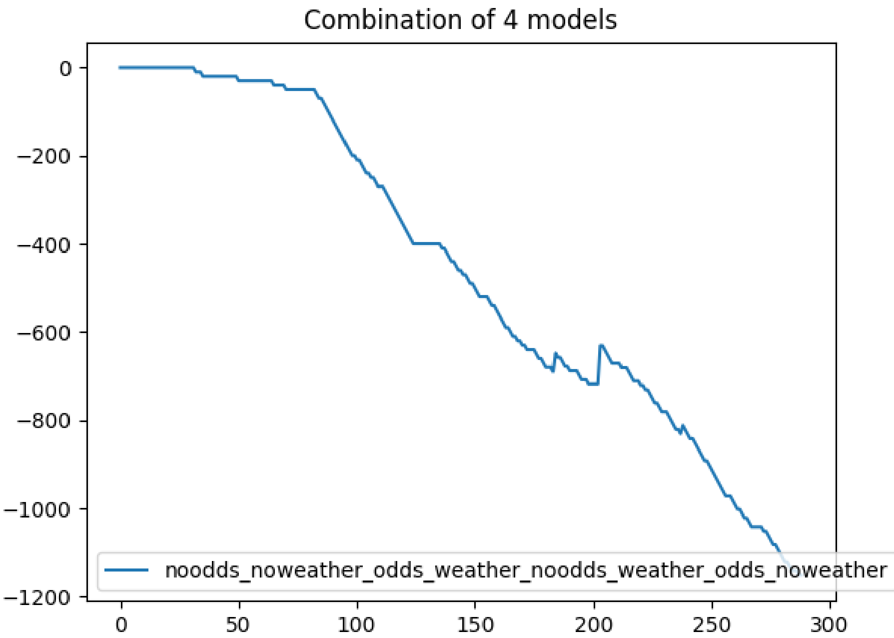


FIGURE 4.12: Net gain of combination of 4 models from divided data set (ST)

As we can observe from the above graphs, the combined models show a good potential in improving the prediction. Overall, the net gain of each set of models rises significantly, and this effect works better in Sha Tin. The best combination so far, combining odds-weather and odds-noweather in Sha Tin, once reached the peak around 1500 HKD of net gain. However as a cost of that, the betting frequency is lowered significantly. For example, for the best combination in Figure 4.5 (odds-weather, odds-noweather and noodds-weather), the combined model only bet 415 times, claiming 55% of the whole number of races.

4.2.3 Excluding special conditions

By examining the prediction provided by the models, we found that sometimes the finishing time they predict has a huge difference. We assume that these cases are abnormal and if the difference is too large, the bets will not be placed.

The following graphs show the net gain of models trained by the same scheme as above, but applying the strategy that if the difference between the predictions made by different models is higher than 5 seconds, this race will be dropped and no bets will be made.

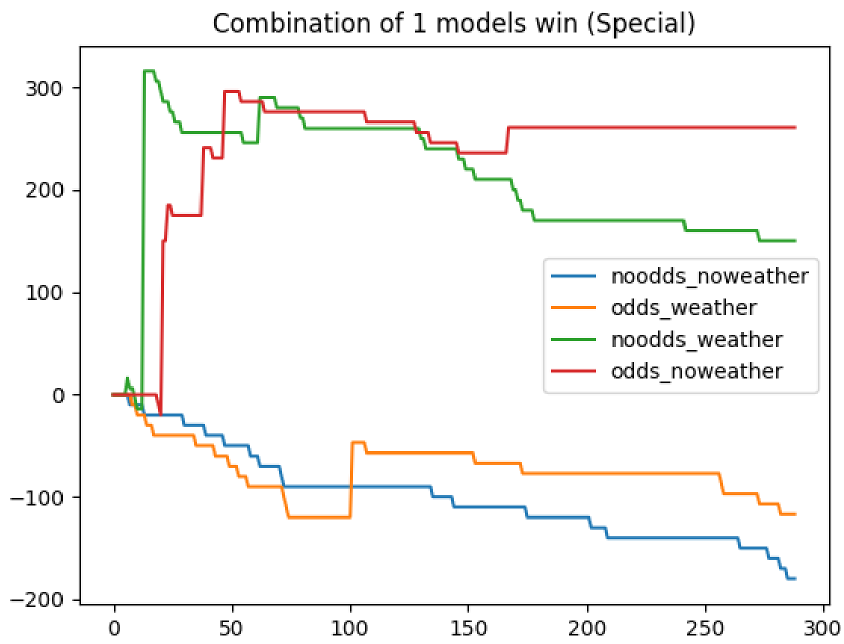


FIGURE 4.13: Net gain of 1 model from divided data set (HV)

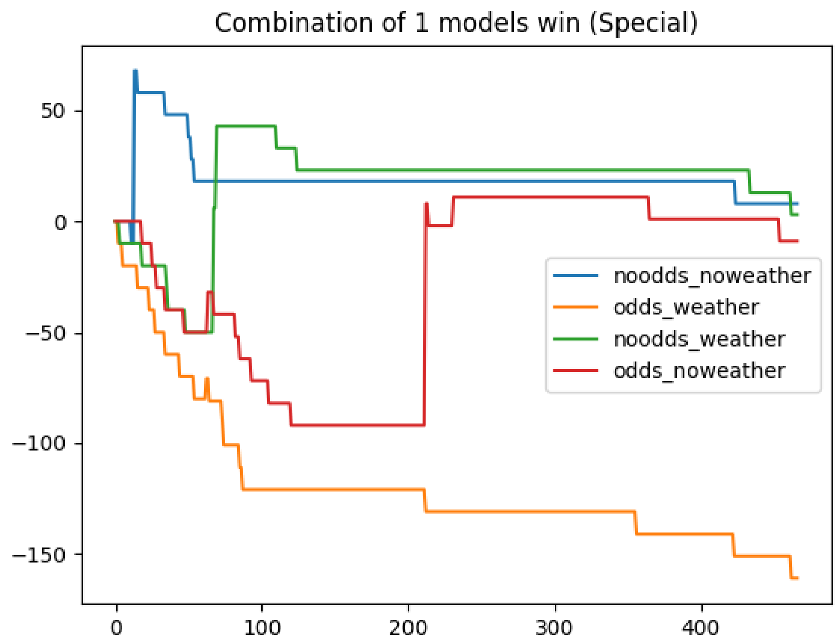


FIGURE 4.14: Net gain of 1 model from divided data set (ST)

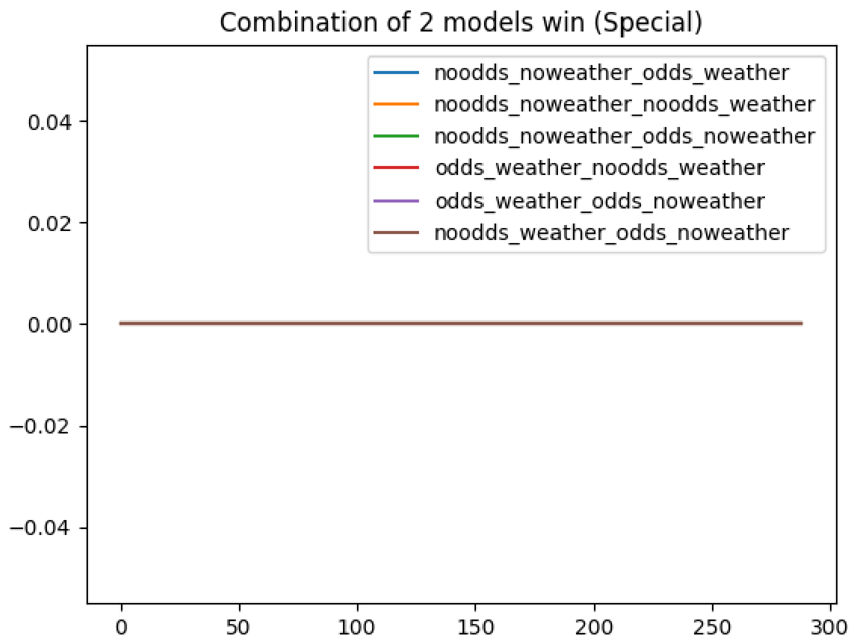


FIGURE 4.15: Net gain of combination of 2 models from divided data set (HV)

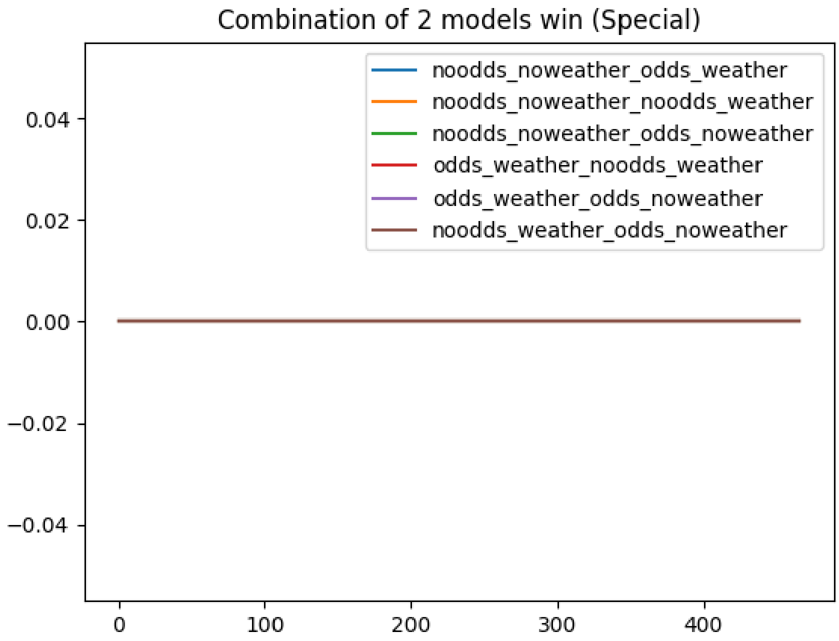


FIGURE 4.16: Net gain of combination of 2 models from divided data set (ST)

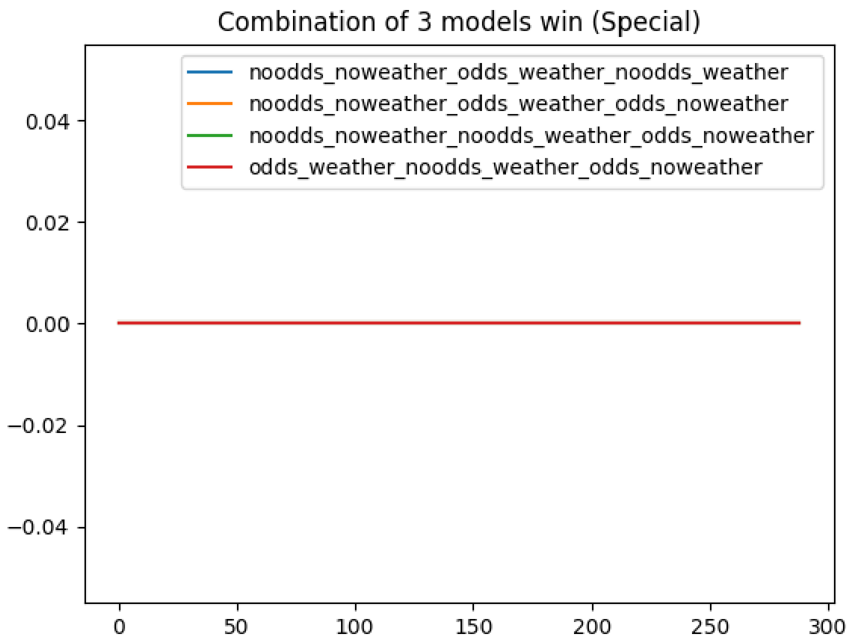


FIGURE 4.17: Net gain of combination of 3 models from divided data set (HV)

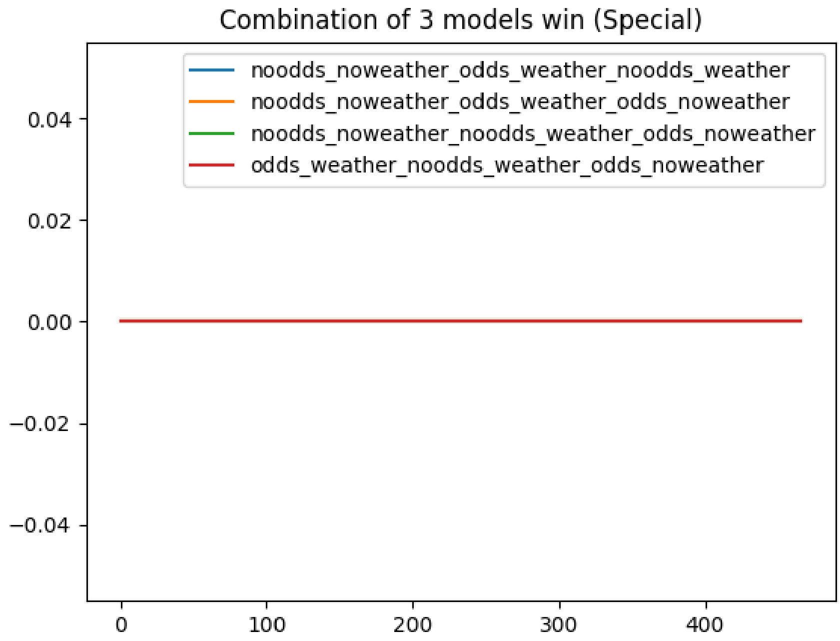


FIGURE 4.18: Net gain of combination of 3 models from divided data set (ST)

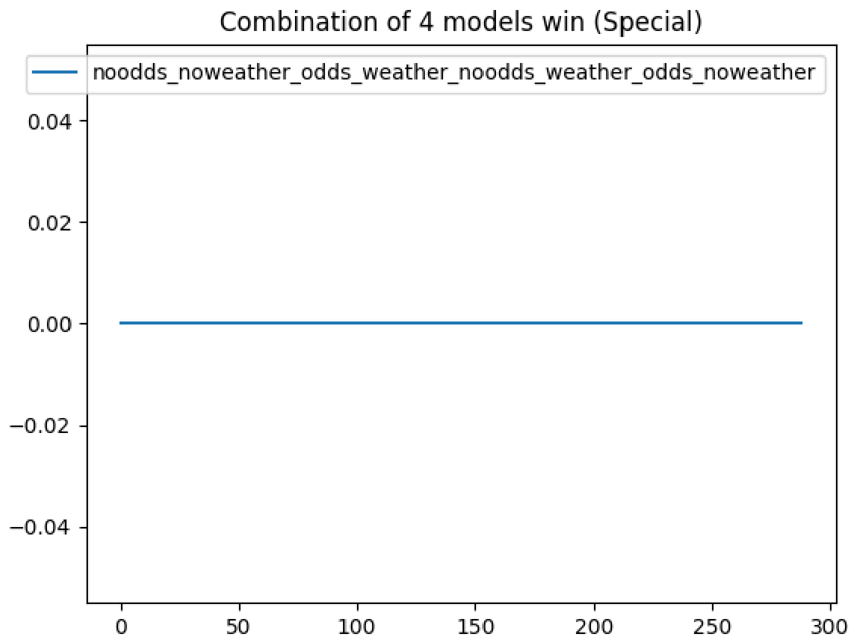


FIGURE 4.19: Net gain of combination of 4 models from divided data set (HV)

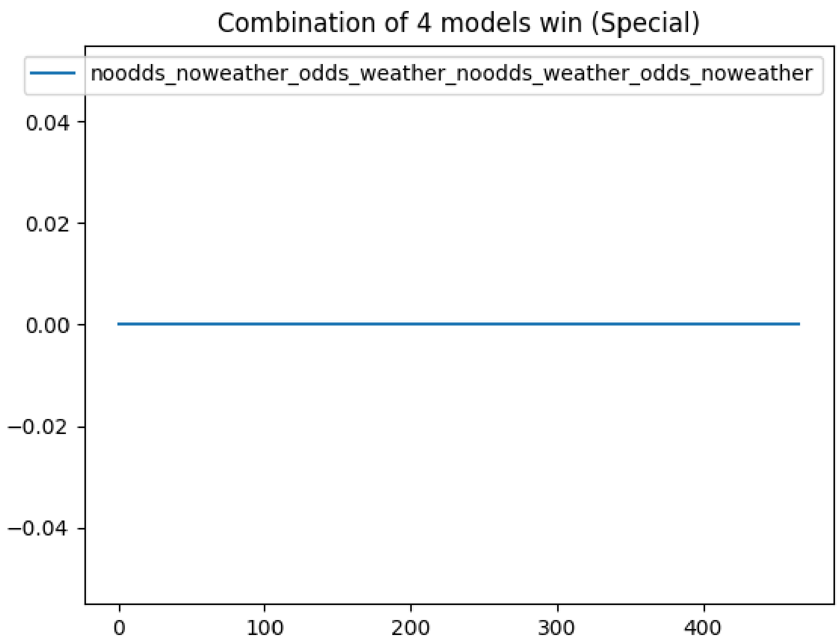


FIGURE 4.20: Net gain of combination of 4 models from divided data set (ST)

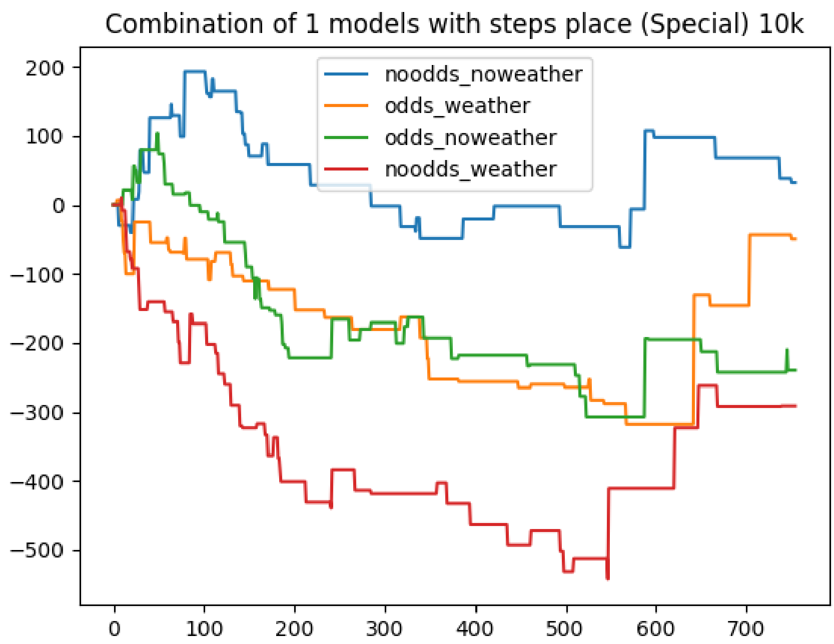


FIGURE 4.21: Net gain of 1 model from undivided data set

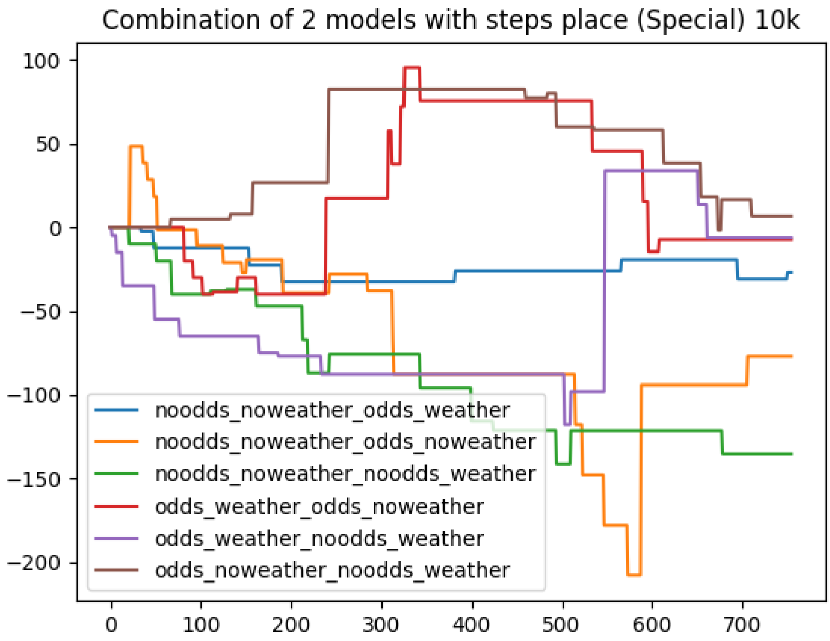


FIGURE 4.22: Net gain of combination of 2 models from undivided data set

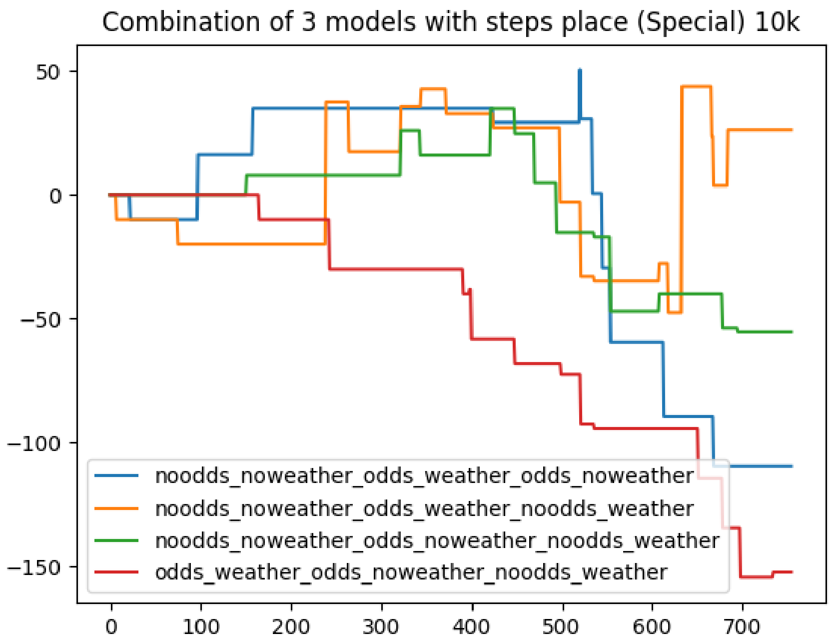


FIGURE 4.23: Net gain of combination of 3 models from undivided data set

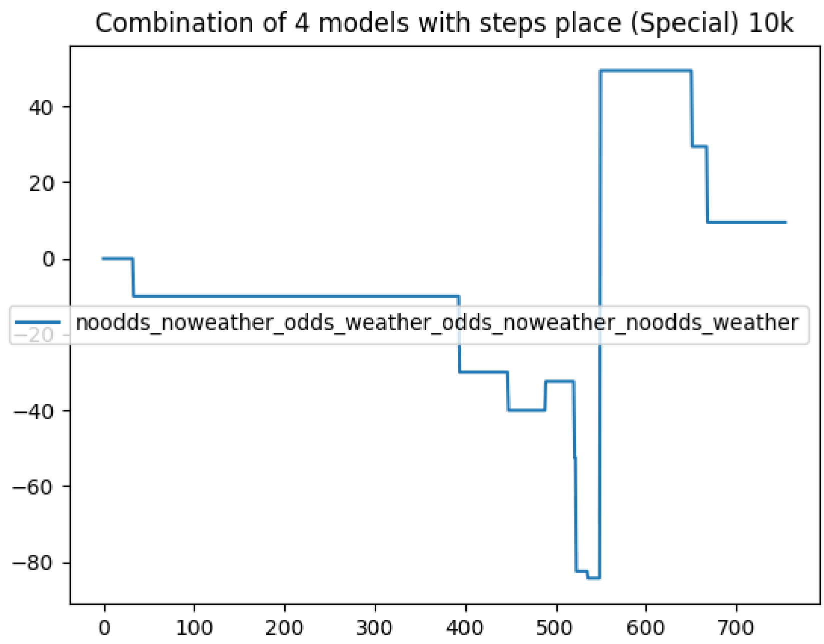


FIGURE 4.24: Net gain of combination of 4 models from undivided data set

The graphs above show that after applying this strategy, the net gain is much more stable. Some of the models can earn money rather steadily. Meanwhile, the drawback of this method is also clear: too few bets are placed. For the combination of 4 models, no bets is placed for the whole year.

4.2.4 Revising the training steps

Just as what we have found previously, the connections between loss, accuracy and net gain are weaker than we expected. However, the best number of training steps was decided simply by the loss of the models, which is no longer solid. As a result of that, we re-conducted some experiments in chapter 3, to check if the configuration can be changed for better. In this part, the experiments are basically the same as the above ones, except the number of training steps is changed to 100k and 1 million.

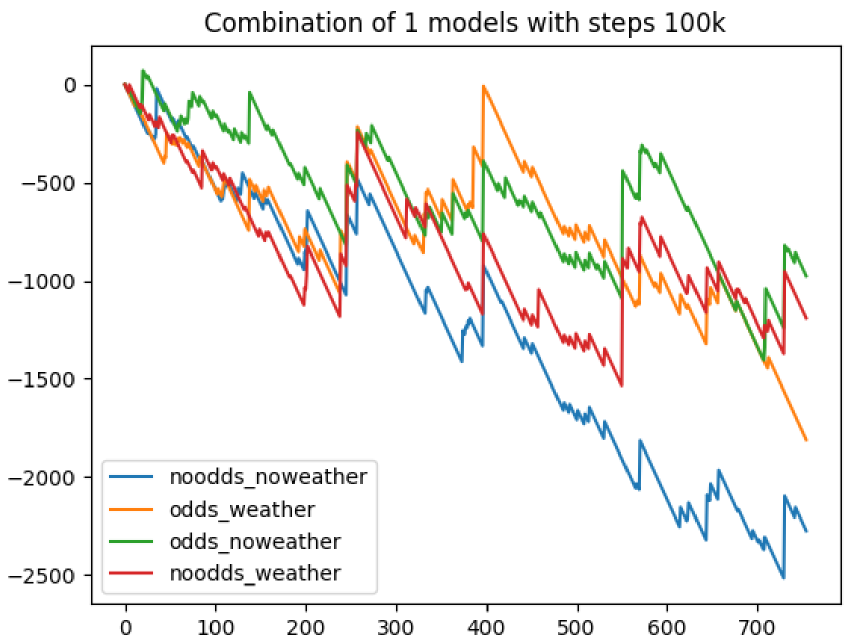


FIGURE 4.25: Net gain of 1 models from undivided data set (100k steps

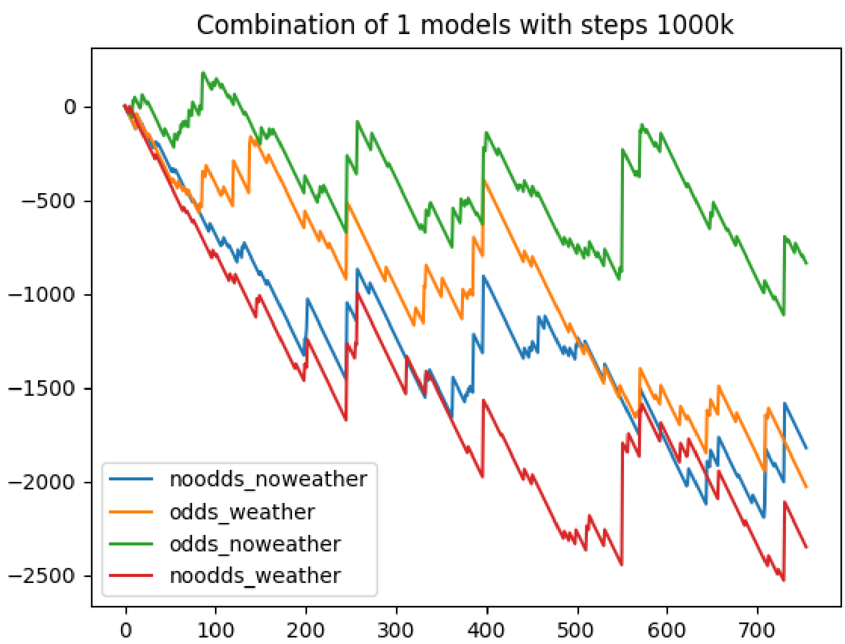


FIGURE 4.26: Net gain of 1 models from undivided data set (1m steps

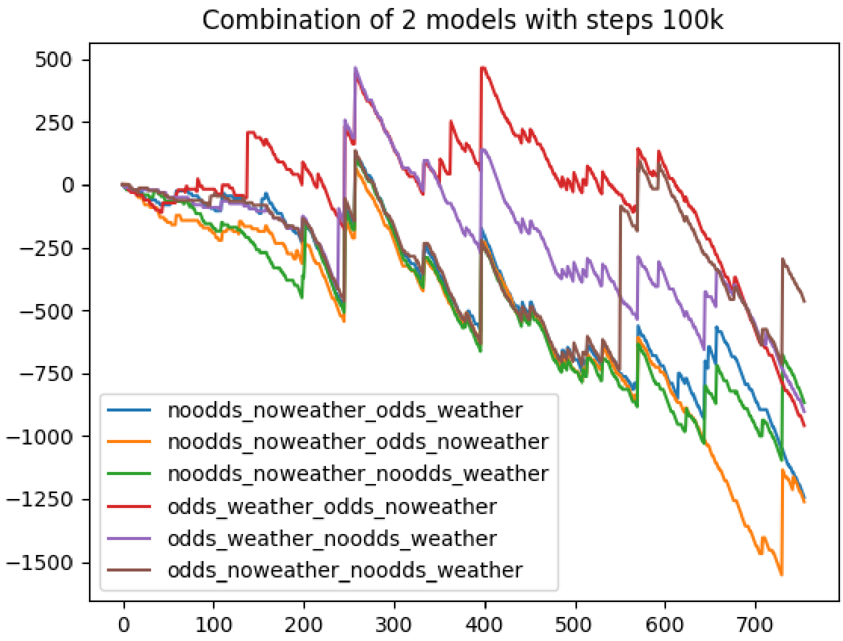


FIGURE 4.27: Net gain of combination of 2 models from undivided data set (100k steps

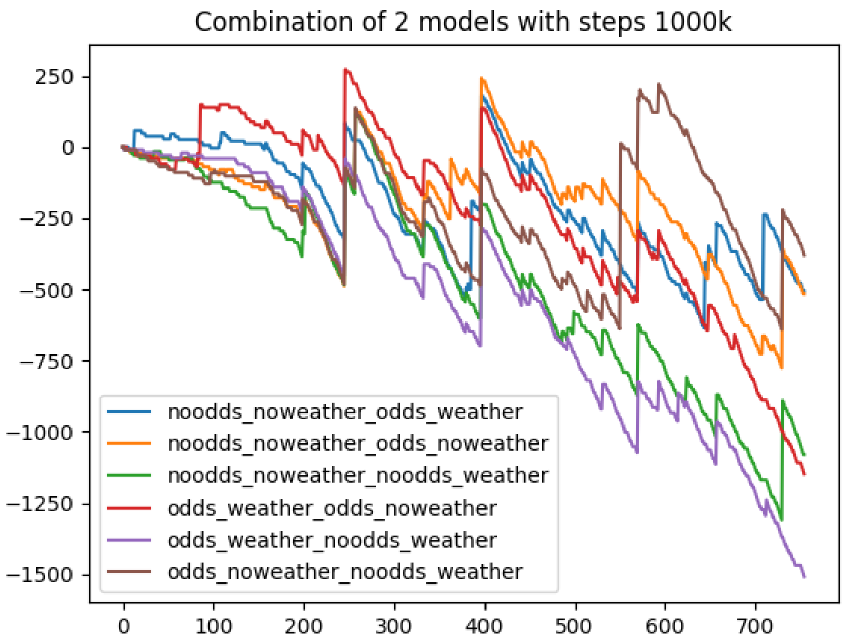


FIGURE 4.28: Net gain of combination of 2 models from undivided data set (1m steps

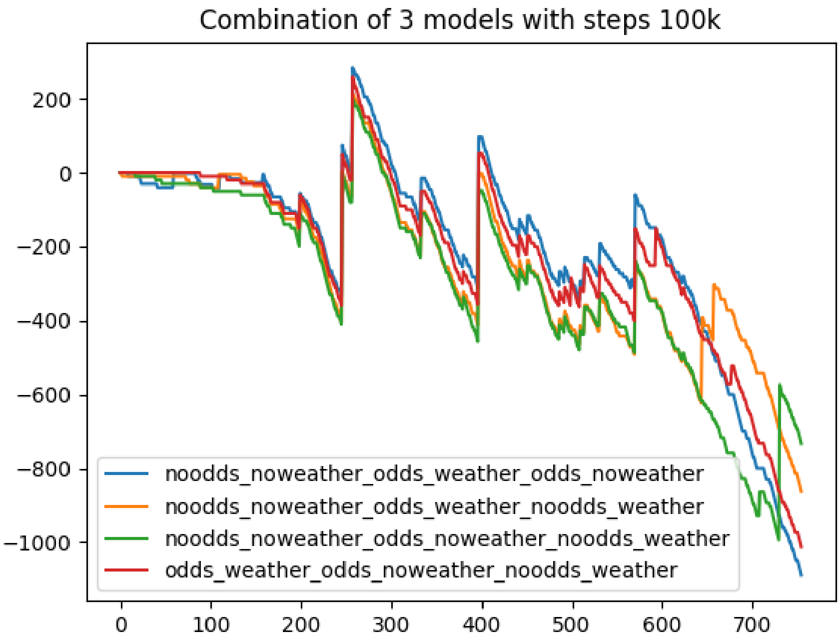


FIGURE 4.29: Net gain of combination of 3 models from undivided data set (100k steps

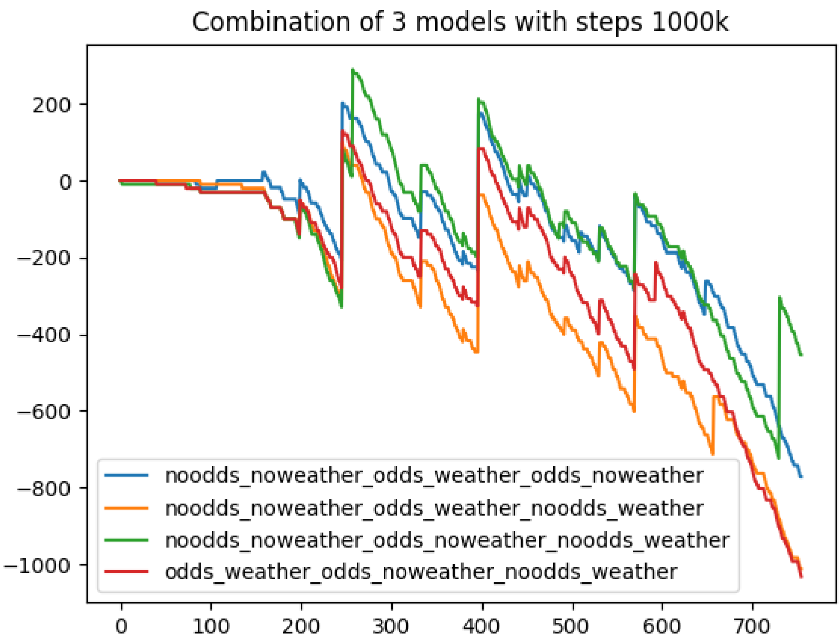


FIGURE 4.30: Net gain of combination of 3 models from undivided data set (1m steps

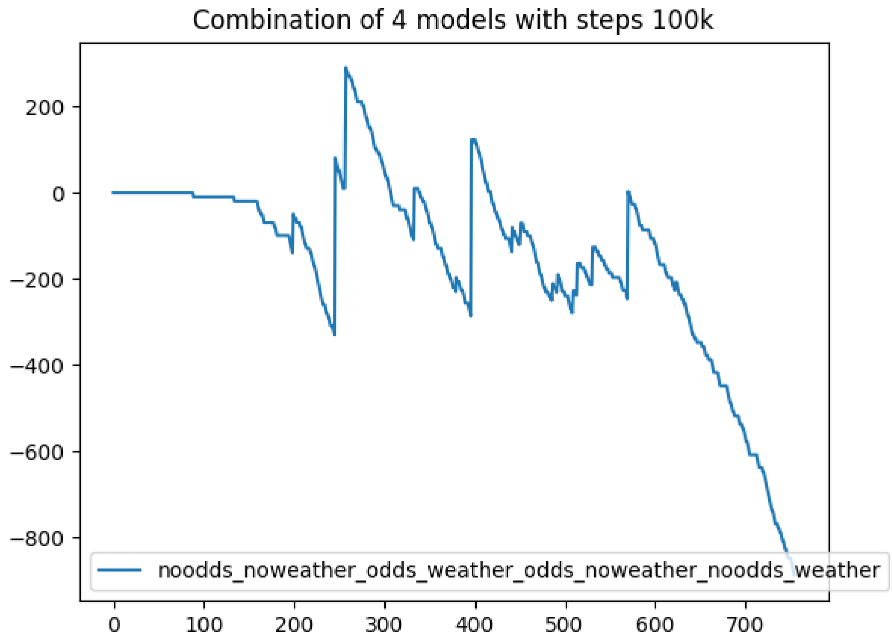


FIGURE 4.31: Net gain of combination of 4 models from undivided data set (100k steps

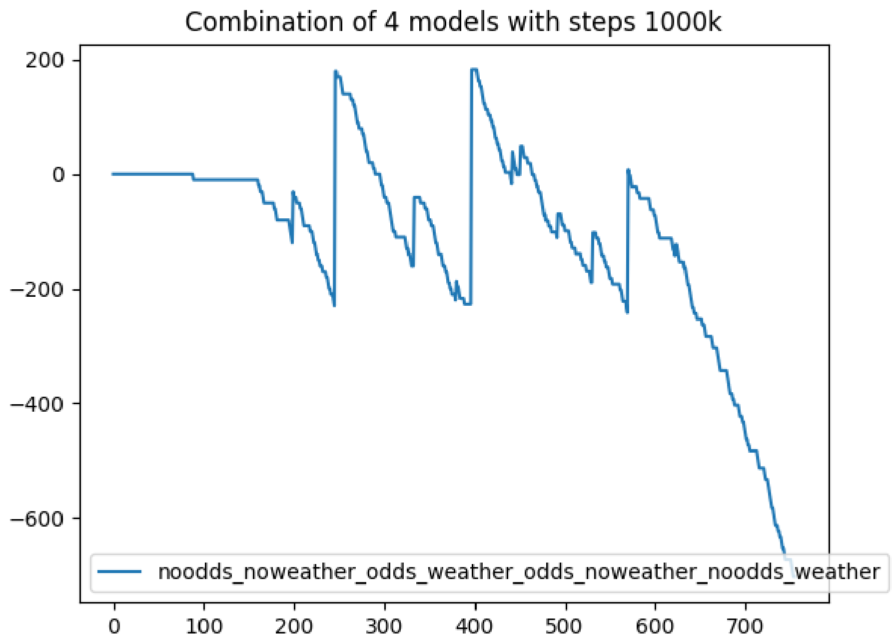


FIGURE 4.32: Net gain of combination of 4 models from undivided data set (1m steps

By comparing models sharing the same feature configuration (including those from the previous section), it can be concluded that, although they are overfitting in terms of loss, 100k and 1m training steps perform even slightly better than 10k training steps with net gains.

4.2.5 Comparison models

Win Bet

The model performed really bad on 2015 – 2016 (Figure 15) and 2016 – 2017 (Figure 16). However, we can agree that the higher the k-value, the better the model will perform, though the order may not be absolute.

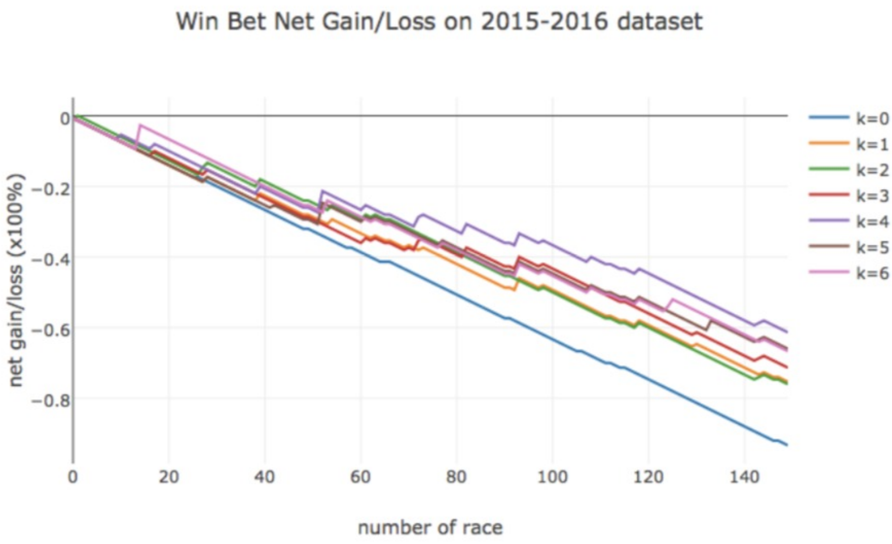


FIGURE 4.33

Comparing with LYU1603, we can tell from the above figure that both $Accuracy_{win}$ and $Accuracy$ are not high enough to obtain positive bet gain.

4.3 Discussion

To interpret the data and figures of the experimental results, we offer some analysis and discussion in this section. The discussion is base on our understandings on horse racing predictions through this semester and will determine our research directions in the future. To facilitate our analysis, we may reuse some of the above tables and introduce figures to help illustrate our ideas.

Model accuracy on finishing time prediction does not guarantee $Accuracy_{win}$ **and** $Accuracy_{place}$**.** While the DNN models are trained to obtain the lowest loss in terms of finishing time of the horses, this does not directly relate to $Accuracy_{win}$ and $Accuracy_{place}$. It leads to a dilemma where models with best loss may still fail in real life.

Training with more features help decrease the test loss but in contrast $Accuracy_{win}$ **and** $Accuracy_{place}$ **may drop.** Because of the similar reasons, there is a gap between the loss and the actual bet accuracy. It implies that the models cannot be justified by loss provided by Tensorflow; it requires more evaluation like accuracy and net gain.

Training more than 10k steps overfits the data sets but $Accuracy_{win}$ **and** $Accuracy_{place}$ **tends to be higher.** We found with surprise that, although the models trained with more steps seem to be overfitting, they generally have a better performance in $Accuracy_{win}$ and $Accuracy_{place}$.

General trends in finishing time matters in horse racing results prediction. Since our model predicts the horse finishing time individually, the predicted finishing time within a race ranges from 1-10.

To better illustrate the issues, the following figure provides the range of horse finishing time with a race. We claims that races with large min-max finishing time are badly-distributed races and the others are normal races.

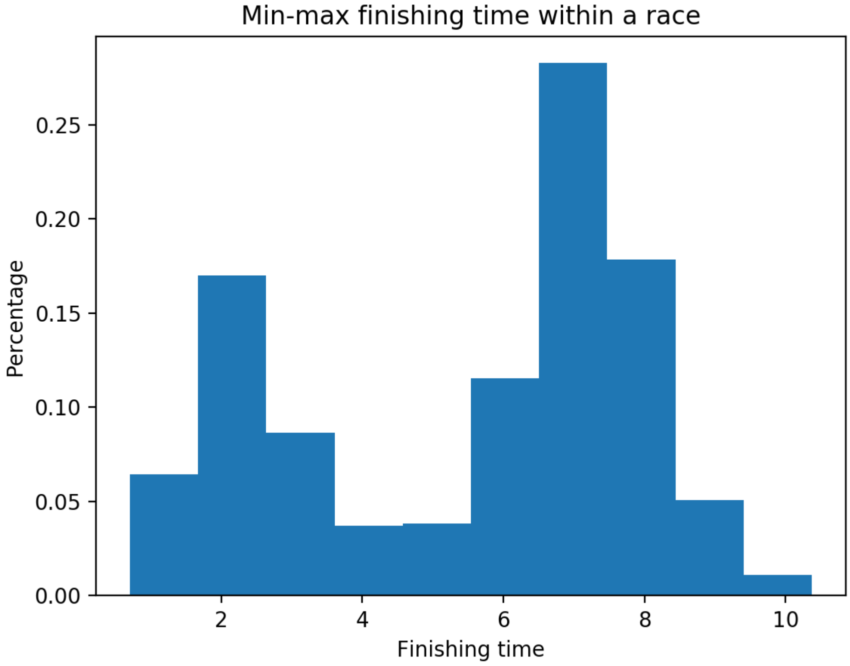


FIGURE 4.34: min-max finishing time distribution of model (1k steps)

Intuitively, $Accuracy_{win}$ and $Accuracy_{place}$ of normal races should outnumber that of badly-distributed races. However, our research shows that $Accuracy_{win}$ and $Accuracy_{place}$ of two kind of races are similar in scale.

Combination of models and Strategies conditions help in horse racing results prediction. We apply two approaches to address the issue and find out both of them are useful in improving $Accuracy_{win}$ and $Accuracy_{place}$.

One approach is to combine models and bet only with high confidence, this approach allow model to "communicate" the finishing time trends to bet on races with high confidence.

Another approach is to bet with strategy. Although we identify that min-max finishing time within a race has little improvement in $Accuracy_{win}$ and $Accuracy_{place}$, combining another strategies focusing on time difference on the first two horses results in a surge in both *Accuracy*. By considering the difference of the first two horses, we hence strictly define what races are regarded as "normal" and the results meet with our understandings.

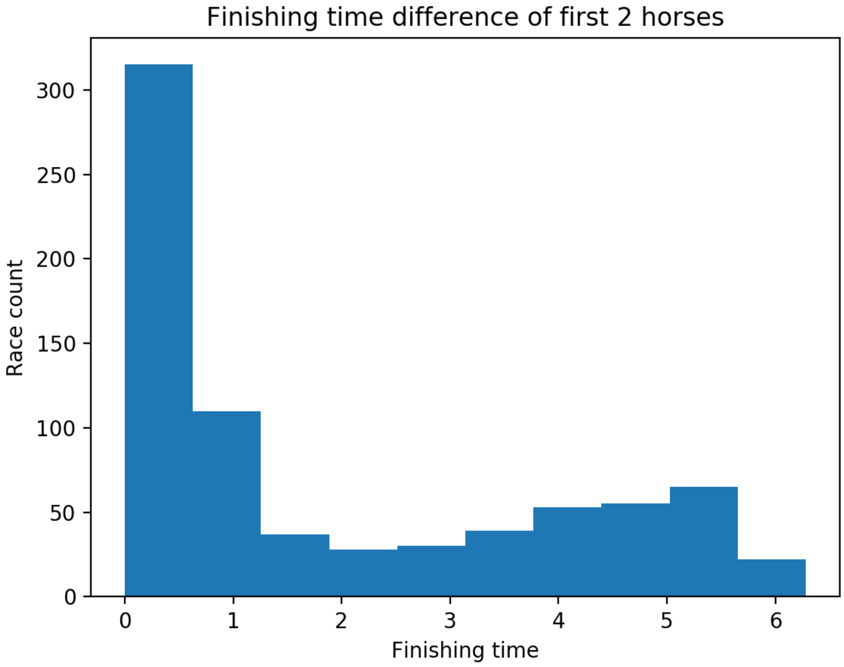


FIGURE 4.35: Finishing time difference of first 2 horses of model (1k steps)

Chapter 5

Conclusion and Future outlook

5.1 Conclusion

This report focuses mostly on discovering the inner-relation among the features, the number of training steps and the predictions. Though a set of experiments, it is clear that predicting horse racing with machine learning is different from most other ML questions in terms of the evaluation of models. Other than the normal "loss" in Tensorflow, the derived prediction accuracy and net gain are also important to evaluate a model. We basically examined three factors, data set division, win odds and weather, and it turned out that races in Sha Tin are significantly more predictable than those in Happy Valley. However, win odds and weather do not show a clear correlation with the final result. We also combined more than one model together and other strategies and got relatively better results, which means this is a correct way to help solve this question.

5.2 Future outlook

In the next semester, we will take a deeper look into the representing following trends of finishing time. A possible way is to group up a race and prediction all finishing time at a time. Another attempt is instead training the average finishing time of a race and design a new loss function to regularize predictions.

Another interesting direction is to train the models on each horses and try to approximate the horse characteristics individually. Either directions discussed is base on our observation of the trends in finishing time. It is believed that once we can train a more accurate finishing time within a race, the higher chance of the trends can be approximated.

Bibliography

Arpit, Devansh et al. (2017). “A closer look at memorization in deep networks”. In: *arXiv preprint arXiv:1706.05394*.

Graves, Alex, Abdel-rahman Mohamed, and Geoffrey Hinton (2013). “Speech recognition with deep recurrent neural networks”. In: *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, pp. 6645–6649.

Harris, David and Sarah Harris (2010). *Digital design and computer architecture*. Morgan Kaufmann.

He, Kaiming et al. (2015). “Deep Residual Learning for Image Recognition”. In: pp. 1 –12. URL: https://arxiv.org/pdf/1512.03385.pdf.

– (2016). “Identity mappings in deep residual networks”. In: *European Conference on Computer Vision*. Springer, pp. 630–645.

Huang, Gao et al. (2016). “Densely Connected Convolutional Networks”. In: URL: https://arxiv.org/pdf/1608.06993.pdf.

Kim, Yoon (2014). “Convolutional Neural Networks for Sentence Classification”. In: URL: http://www.aclweb.org/anthology/D14-1181.

Lange, Sascha and Martin Riedmiller (2010). “Deep Auto-Encoder Neural Networks in Reinforcement Learning”. In: URL: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=5596468.

LeCun, Yann et al. (1998). “Gradient-Based Learning Applied to Document Recognition”. In: *PROC. OF THE IEEE*, pp. 1 –46. URL: http://yann.lecun.com/exdb/publis/pdf/lecun-01a.pdf.

Méhat, Jean and Tristan Cazenave (2011). “A parallel general game player”. In: *KI-künstliche Intelligenz* 25.1, pp. 43–47.

Ouyang, Xi et al. (2015). “Sentiment Analysis Using Convolutional Neural Network”. In: URL: http://ieeexplore.ieee.org/document/7363395/.

Ren, Shaoqing et al. (2015). “Faster R-CNN: Towards Real-Time Object Detectionwith Region Proposal Networks”. In: URL: https://papers.nips.cc/paper/5638-faster-r-cnn-towards-real-time-object-detection-with-region-proposal-networks.pdf.

Riess, Steven A (1991). *City games: The evolution of American urban society and the rise of sports*. Vol. 114. University of Illinois Press.

Santos, Cicero Nogueira dos and Maira Gatti (2014). "Deep Convolutional Neural Networks for Sentiment Analysis of Short Texts". In: URL: http://www.aclweb.org/anthology/C14-1008.

Silver, David et al. (2016). "Mastering the game of Go with deep neural networks and tree search". In: *Nature* 529.7587, pp. 484–489.

Snyder, Wayne (1978a). "Decision-making with risk and uncertainty: The case of horse racing". In: *The American Journal of Psychology*, pp. 201–209.

Snyder, Wayne W (1978b). "Horse racing: Testing the efficient markets model". In: *The Journal of finance* 33.4, pp. 1109–1118.

Sola, J and J Sevilla (1997). "Importance of input data normalization for the application of neural networks to complex industrial problems". In: *IEEE Transactions on Nuclear Science* 44.3, pp. 1464–1468.

Srivastava, Nitish et al. (1989). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Neural Networks* 2.15, pp. 359 –366. URL: https://www.cs.cmu.edu/~epxing/Class/10715/reading/Kornick_et_al.pdf.

– (2014). "Dropout: A Simple Way to Prevent Neural Networks from Overfitting". In: *Journal of Machine Learning Research* 15.15, pp. 1929 –1958. URL: https://www.cs.toronto.edu/~hinton/absps/JMLRdropout.pdf.

Srivastava, Rupesh Kumar, Klaus Greff, and Jurgen Schmidhuber (2015). "Deep Residual Learning for Image Recognition". In: pp. 1 –6. URL: https://arxiv.org/abs/1505.00387.

Szegedy, Christian, Alexander Toshev, and Dumitru Erhan (2016). "Deep Neural Networks for Object Detection". In: URL: https://papers.nips.cc/paper/5207-deep-neural-networks-for-object-detection.pdf.

Tung, Cheng Tsz and Lau Ming Hei (2016). "Predicting Horse Racing Result Using TensorFlow". In: URL: http://www.cse.cuhk.edu.hk/lyu/_media/students/lyu1603_term_1_report.pdf?id=students%3Afyp&cache=cache.

Vanhoucke, Vincent, Andrew Senior, and Mark Z. Mao (2010). "Improving the speed of neural networks on CPUs". In: URL: https://static.googleusercontent.com/media/research.google.com/zh-CN//pubs/archive/37631.pdf.

Widrow, Bernard and Marcian Hoff (1959). "Adaptive Switching Circuits". In: URL: http://www-isl.stanford.edu/~widrow/papers/c1960adaptiveswitching.pdf.

Williams, Janett and Yan Li (2008). "A case study using neural networks algorithms: horse racing predictions in Jamaica". In: *Proceedings of the International Conference on Artificial Intelligence (ICAI 2008)*. CSREA Press, pp. 16–22.

Zhang, Xiang, Junbo Zhao, and Yann LeCun (2015). "Character-level Convolutional Networks for Text Classification". In: URL: https://arxiv.org/pdf/1509.01626.pdf.