Machine Learning with Trustworthy AI for Cycling Race Prediction in Sports Betting

Ray-I Chang[1]*, Huang Ching Liu[1]

[1]Department of Engineering Science & Ocean Engineering, National Taiwan University, Taipei, Taiwan

** Corresponding author: rayichang@ntu.edu.tw*

Abstract

In today's era of booming AI, machine learning (ML) has reshaped how we analyze sports events. Cycling race prediction for sports betting is one of them. This paper establishes cycling betting in an online gambling system. The user can give a stake, and the system will bet according to the prediction result of ML. Finally, an output amount is provided to measure the profit. Notably, conventional ML provides only the AI model without considering its trust in prediction. However, in betting/gambling, it is crucial to find an indicator for measuring the trust of the AI model in its prediction. In this paper, we design an algorithm that can evaluate the trust of the AI model efficiently and effectively. Our experiments use the rider data selected from 2014 to 2018 for ML, and the predicted target is the top10 in 2019 Tour *de* France. The experimental results show that the precision rate of the top10 obtained by the basic AI model can reach 62%. Then, we try to add our trust score in prediction, and apply only the trusted AI prediction in betting. Results show that the maximum profit can increase by about 40%.

Keywords: Cycling, Data Analytics, Sports Betting, Machine Learning, Trustworthy AI

1. Introduction

Due to the increasingly convenient collection of sports data, sports analysis has emerged in recent years. The direction of earlier research is more popular in ball sports, such as football, basketball, and rugby. In contrast to cycling races, competition did not appear until the 1900s (Gabriele, 2011), so there are fewer studies analyzing cycling events. Recently, there are some sports analysis studies using AI (artificial intelligence) and machine learning (ML). Unlike the manual recording and analysis of data, the use of ML can more effectively analyze data, and improve sports analysis to a new level (Soneja, 2019). Athletes can train and improve themselves through the results of sports analysis, and choose the best strategy in the competition. Broadcasters, able to broadcast game statistics in real-time through sports analysis. Spectators, who can participate in sports betting through sports analytics. This paper studies the part of spectators participating in sports betting, and tries to apply ML to predict the outcome of cycling events and

improve profitability. Since sports betting itself is a game with probability components, how to judge whether the prediction result of the AI model is trustworthy is an important topic (Das, 2019). Therefore, this paper proposes an innovative trust reference, so that users can use this trust to create maximum profit.

Sports betting has a long history (Gilchrist, 2020). It can be traced back to Greece more than 2,000 years ago. It was turned to underground betting due to religious factors. Nowadays, the emergence of horse racing betting let sports betting become legal and spread rapidly around the world. Therefore, betting on various events also appeared one after another. Compared with ball sports betting, although cycling betting appeared later, it is still quite popular. In this paper, we apply ML to build a cycling betting and reward prediction system, which we can predict the top10 of Tour *de* France and finally input stake X dollars to give profit Y dollars. The simple architecture is as follows: first, ML method is used to predict whether the rider is in the top10, and then the final profit is calculated according to the predicted result. In addition, we put the prediction results into the trust calculation module, and finally adjust our betting combination according to the trust score to output the optimized final profit. In summary, the main contributions of this paper are listed as follows.

1. Build a prediction system for cycling betting.
2. Propose an algorithm for trustworthy AI.
3. Use trust to optimize profit.

2. Related works

The types of sports that use AI for sports analysis are dazzling, and the research objectives and the results of interest are also different. In (Thabtah, 2019), different AI models were used to predict the outcome of the game through the data recorded by the USA Basketball Association, and they compared the performance of the model the proposed, furthermore, they also found the most significant feature influencing the results. The study (David, 2011) used official data recorded by the American Football League and used neural networks to predict the results of American football. The model is built using statistical differentials to compare teams, and finally through principal component analysis to determine which statistics has the greatest impact on the model. In baseball, (Valero, 2016) predicted the final result based on the ranking of teams and players, and evaluated four algorithms they used in the works.

In the field of cycling, the main focus of research is to predict the relevant race data or to study the results of the race. Among the relevant race data, there are two popular ones, namely heart rate prediction and power performance prediction. Heart rate and

power performance have always been the data that professional riders pay attention to. They can make training more scientific and can monitor training intensity in real time. And influence the trend of the entire event, benefiting the riders and coaches. In (Mutijarsa, 2016), neural network (NN) was used to create a heart rate prediction model for riders. After comparing with the original sensor data, the final error value was 2.43 for the training data and 3.02 for the test data. In (Kataoka & Gray, 2018), they used GPS data to design and build a real-time predicted power output system for Tour *de* France. Their features were generated by autoencoder and hand-selected. Finally, the Mean Absolute Error (MAE) of AI model is lower than the conventional physics model. It is helpful to a certain extent in training riders, and riders can use these data to improve their ability.

Although there are few studies on the prediction of race results, there are still many studies worthy of reference. The study (De Spiegeleer, 2019) uses data from past races to predict the outcome of future races. This study focuses on predicting the outcome data of 3 races, namely the average velocity of a stage, the difference between the average stage velocity and the velocity of a rider, and the number of wins in head-to-head matches between two riders in a stage. It uses data from past races to predict the outcome of future races. This study focuses on predicting the outcome data of 3 races, namely the average velocity of a stage, the difference between the average stage velocity and the velocity of a rider, and the number of wins in head-to-head matches between two riders in a stage. (Kholkine, 2020) used historical data from the past to predict the final results of bicycle races, they used XGBoost as the algorithm to build the whole architecture, and finally predicted the top10 rider rankings of Tour of Flanders. Although research above can predict the final game result, but Tour of Flanders is a one-day game, unfortunately it cannot predict the result of the multi-day game and lacks substantial application.

In (Logg et al., 2018), it concluded that people often choose their own decisions between the results of the algorithm and their own decisions, because most people still have low trust in the algorithm. It's important to build a trustworthy AI model. The study (Lee, 2021) organizes various data about trust in AI model. (Hoff & Bashir, 2015) divided trust into three types: dispositional trust, situational trust and learned trust. Among the dispositional trust, there are some attributes will mainly affect their trust, such as culture, age and gender, etc. Situational trust includes professional knowledge, risk, and mood, etc. Learned trust can be divided into initial learned trust and dynamic learned trust. Dynamic learned trust is particularly important, users will become more familiar with this model if they use it repeatedly, which will increase dynamic learned trust. Therefore, the transparency of the model is particularly important. If it is easier for users to understand the internals of the model, the trust will increase. In (Thiebes, 2021),

the concept of trustworthy AI and its five basic principles are introduced, namely beneficence, non-maleficence, autonomy, justice and explicability. The explicability corresponds to the transparency of the model (Hoff & Bashir, 2015). (Samek, 2017) gives a definition in the AI of explicability, that is, the methods and technologies that allow experts to understand the achievements of AI, and classify them into the following aspects, namely verification of the system, improvement of the system, learning from the system and compliance to legislation.

3. Method

This paper proposes a trustworthy AI strategy, which allows users to have a reference direction in gambling applications, so as to make trustworthy investments and bets. We have designed a complete betting system, which combines the prediction of the top10 in cycling events, betting profit prediction, and trust calculation and optimization, as shown in Figure 1. The input part is a stake, and then after our ML module, it is divided into two parts, one part will output our profit directly through the odds calculator, the other part will enter our trust module to calculate the trust of our model, and finally according to the score of the trust, the system will redistribute the stake, and finally calculate the optimized profit.

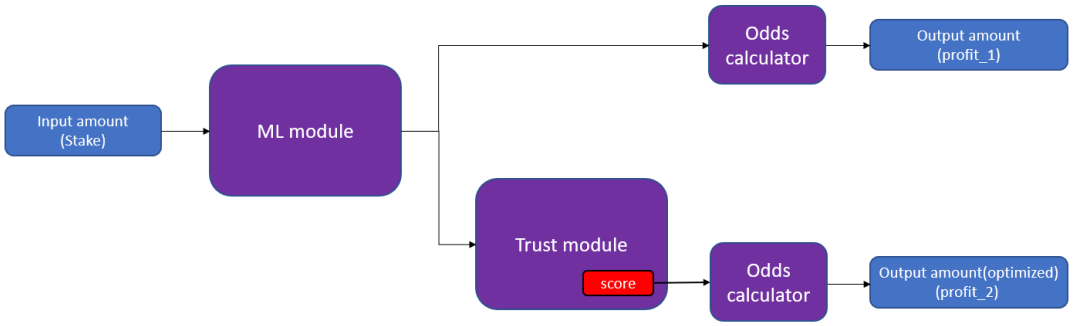

Figure 1 System structure

The data source for this study is from https://cqranking.com/, which contains information on many riders. The ranking calculation benchmark used is CQ ranking, and CQ scores are used to calculate rankings and so on. The preliminarily captured data is divided into two parts. The first part is the basic information of each rider, and the second part is all the events that each rider participated in. As this study made predictions for Tour *de* France, only the teams qualified to participate in Tour *de* France were selected, and the data of a total of 18 teams were selected. Through the basic information of the riders extracted above and the participation of each rider, 17 features were extracted and

all features are then normalized to form a Gaussian distribution. According to the betting rules, the profiles were labeled as whether they belonged to the top10 riders or not (0=NO and 1=YES).

These 17 features are listed as follows: (1) rank_start, (2) point_start, (3) win, (4) podium, (5) top_ten, (6) points, (7) race_days, (8) win_dauphine, (9) podium_dauphine, (10) top_ten_dauphine, (11) points_dauphine, (12) win_swiss, (13) podium_swiss, (14) top_ten_swiss, (15) points_swiss, (16) climb, (17) sprint. In the pre-processing part, 17 different features are mentioned, which can be roughly divided into three categories.

D1: the basic data of the riders (features (1) to (7)).

D2: the related races of Tour *de* France (features (8) to (15)).

D3: the riders' specialization (features (16) to (17)).

The first category of data *D1* of the riders is based on the statistics of each rider's past competition data. The second category of data *D2* considers the statistics of races that are highly related to Tour *de* France to improve the impact on the overall training results. In (Sheehan, 2018), it shows that Criterium *du* Dauphine ($D2.Dau$) and Tour *de* Suisse ($D2.Sui$) are often regarded by riders as the pre-race training of Tour *de* France. The correlation among these races, whether from the schedule and terrain, or past data, shows the importance of these two races to Tour *de* France (Lowe, 2016). In the third category of data *D3* are riders' specialization, namely climbing ($D3.climb$) and sprint points ($D3.sprint$).

Among these three categories of data, we try to find the best combination to get the best training results. Five data sets for training are given as follows.

$$Train_1 = D1 + D2.Dau + D3.climb + D3.sprint \quad (1)$$

$$Train_2 = D1 + + D3.climb + D3.sprint \quad (2)$$

$$Train_3 = D1 + D2.Dau + D3.climb \quad (3)$$

$$Train_4 = D1 + D2.Dau + D3.sprint \quad (4)$$

$$Train_5 = D1 + D2 + D3 \quad (5)$$

The following describes the process and algorithm for calculating the trust in sequence. The central idea of this research is that the closer the predicted input and output are to the input and output of learning samples, the more credible the prediction is. As shown in Figure 2, we use SMOTE (Synthetic Minority Oversampling Technique) to balance our data for the test data (Test_X) and the prediction data (pred_Y) output by the ML module. SMOTE is a synthetic data algorithm based on comprehensive sampling, which is used to solve the problem of data imbalance. Because in this study, there are fewer labels in the top10, SMOTE is chosen to improve our data, and then input to the KNN (K-Nearest Neighbor) model for fitting, and then use the original learning samples (Train_X) to predict the trained KNN, the above actions are equivalent to finding

matching points in the distribution generated by the test data (Test_X) and the prediction data (pred_Y) output by the ML module, and finally output the prediction data (KNN_predict), so we can compare this data with the original learning sample.

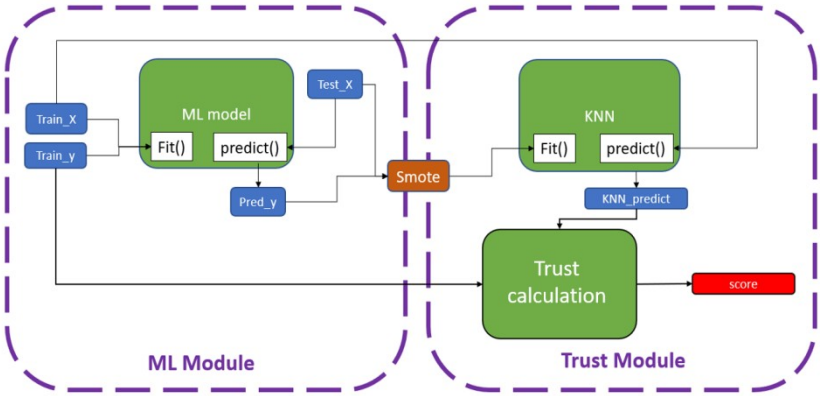

Figure 2 Detailed content of ML module and Trust module

Defining and calculating the similarity of the data will be of great help to the calculation of the trust later. Therefore, a suitable metric must be selected to measure and help to judge the difference of the data, and we finally choose to use the Euclidean distance to carry out the experiment, assuming that in the Euclidean space, the equations solved for point x = $(x_1,\ldots,x_n)$ and point y = $(y_1,\ldots,y_n)$ Euclidean distance can be expressed as follows.

$$d(x,y)=\sqrt{\sum_{k=1}^{n}(x_k-y_k)^2} \tag{6}$$

We calculate each distance d between the prediction and training data and then use the following formula.

$$Trust=\frac{1}{(1+d)} \tag{7}$$

As our similarity, the calculated range is between 0 and 1, the closer it is to 1, the more similar it is, and the closer it is to 0, the less similar it is. We first compare the results predicted by KNN with the training sample data (train_y), and then divide it into two parts. If the prediction is correct, it means that the trust will be higher. At this time, the distance is selected according to the predicted label. If the prediction is incorrect, representing a low trust, a distance must be given to punish, so theoretically, the distance must be selected according to the label value opposite to the prediction, but there will be some special cases, such as the shortest penalty distance, in this case, the average of several other distances must be chosen for the calculation.

The flow chart is shown in Figure 3. Assume that (d1,d2,d3)represent a certain distance which is the results output by a function *Kneighbors*(train_X, n_neighbors=3).

First, we first judge whether the results predicted by KNN are equal to the data of the original training samples (train_y). If they are equal, we will find the corresponding y value in (d1,d2,d3) according to the result predicted by KNN, finally select all the found distances, and take the average and throw it into the similarity calculation to calculate the final score.

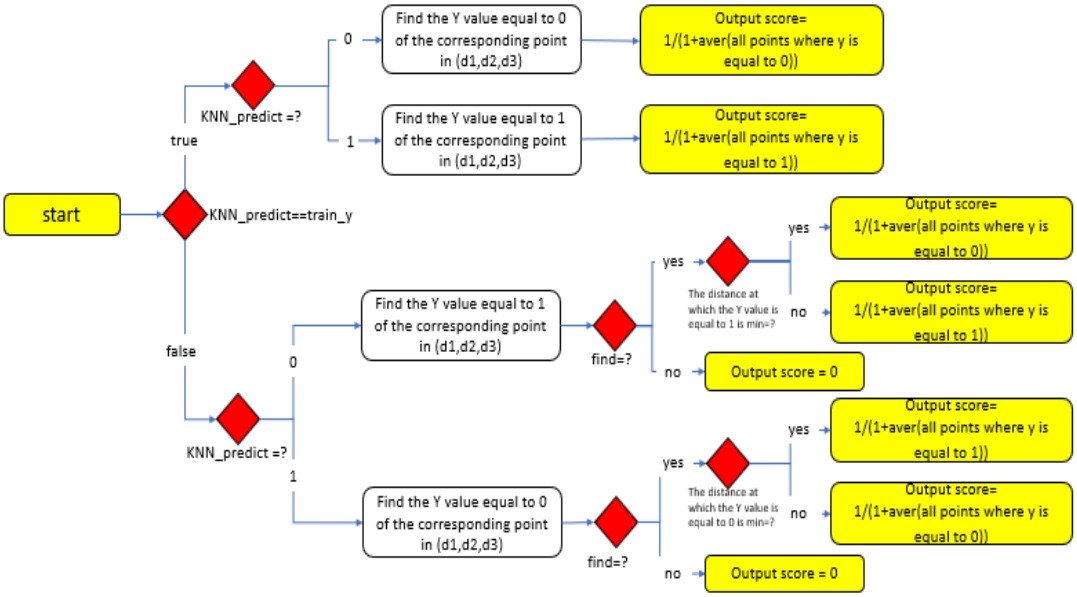

Figure 3 Trust calculation flow chart

As shown in Figure 4(a), KNN_predict and train_yare equal at this time, and the distances represented by d1, d2 and d3 are as follows. We select the average of d1 and d2 to bring into the similarity calculation formula for calculation, because the labels represented by these two distances are both same as KNN_predict. Let the function *aver*(d1, d2) be the average of the two, the similarity score can be expressed as follows.

$$Trust = \frac{1}{1 + aver\left(d_1, d_2\right)} \tag{8}$$

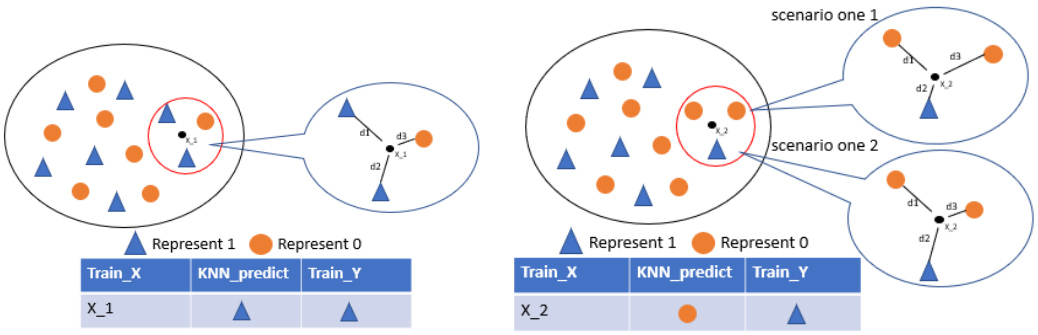

Figure 4 Data distribution of two special cases (a) and (b)

The following is the case where the result predicted by KNN is not equal to the data of the original training sample. As in the case of equality, first determine that the result predicted by KNN is 0 or 1. If it is equal to 0, we will look for the distances represented by label 1 in (d1, d2, d3), if not found, the trust score is 0. If it is found, it is judged whether the distance represented by the label 1 in (d1, d2, d3) is smaller than the other distances. If it is smallest in three distances, the average of all the distances represented by label 0 is taken into the similarity formula for calculation, otherwise, the average of all the distances represented by label 1 is taken into the similarity formula for calculation. As shown in Figure 4(b), in scenario one, because KNN_predict is equal to 0, and d2 is smaller than d1 and d3, we choose the average of d1 and d3 to be added to the similarity formula calculation, and the similarity can be expressed as follows.

$$Trust = \frac{1}{1 + aver\left(d_1, d_3\right)} \tag{9}$$

But in scenario two, d2 is not the smallest distance, so we only select d2 and bring it into the similarity formula for calculation.

$$Trust = \frac{1}{1 + d_2} \tag{10}$$

This part is the final part of the whole system. The above results are passed through the odds calculator to output the final amount of whether to add the trust score. The following is divided into odds calculation and final output algorithm.

According to the conception of this study, the ultimate goal is to predict the top10 riders of Tour *de* France, and finally make bets. In order to calculate the final output amount, it is a necessary step to simulate the odds of each rider. According to the above, the odds can be expressed as the following formula.

$$Odd = \frac{1}{probabilty\ of\ top\ ten} \tag{11}$$

That is, it is inversely proportional to the probability of top10. Then we apply the activation function contained in the above model to simulate the odds, because through the activation function, we can see the probability of top10 and non-top10, and finally add the range of odds that can be set by ourselves, which can be expressed as the following formula.

$$Odds = \left(\frac{1}{P} - 1\right) \times Odds_{range} \tag{12}$$

P is the probability of top10, that is, the probability output by the activation function, and Odds_range is the range of odds that can be set by ourselves. For example, if Odds_range is set to 10 in this study, the output range of the odds calculation will fall

between from 1 to 10.

There are two cases in this part, the amount output without adding the trust score and the amount output with adding the trust score, which will be discussed separately below.

(1) Without adding the trust score. Combine the results of the ML module with the odds calculator to get at the final amount.

(2) Adding the trust score. This part is the amount output by adding the trust score.

We cooperate with the trust degree to adjust the stake of the rider who needs to bet predicted by the ML module, so that the rider with low odds bet a little more money, and vice versa. The main reason is to adopt a conservative strategy to increase the money we have won and reduce the possible loss. In order to choose which riders need to raise or lower their stakes, we set an odds benchmark, and we will reduce the capital if the bettor's odds are lower than this benchmark, and increase the capital otherwise. The odds benchmark is equivalent to the amount of risk taken, and above this standard is equivalent to an increase in risk and therefore a reduction in the stake, and vice versa. We use trust to calculate the odds benchmark, the formula is as follows

$$Odds\,benchmark = (Trust * W) \times Odds_{range} \tag{13}$$

Multiply the previously calculated trust score by W (Trust Weight) and finally multiply by Odds_range. Through the above algorithm, the range of the odds benchmark can be set to be the same as the odds calculated in the previous step.

Finally, the final output amount is calculated according to the following process, as shown in Figure 5. It is mainly to determine whether the estimated odds of each data to be bet are lower than the odds benchmark. If it is lower, increase the stake, otherwise decrease the stake. After setting the amount of capital increase and decrease, you can use the results predicted by the ML module to bet, and finally the final output amount can be obtained.

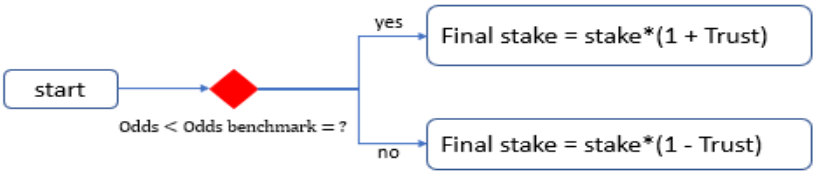

Figure 5 Flow chart of the stake after adding the trust reference

4. Experimental results

This section presents the experimental results of this study, and introduces our training and testing data, as well as the input and output of the entire cycling gambling

system in sequence. The data from 2014 to 2018 were selected as training and testing data, with a total of 648 data. There are two types of labels: riders who belong to the top10 of Tour *de* France and riders who do not belong to the top10 of Tour *de* France. The sample numbers of the top10 riders and the non-top10 riders are 50 and 598 respectively. The final split ratio of training dataset and test dataset is 2:1. We predict the outcome of Tour *de* France from 2019 to 2021. The input and output of this experiment are all amounts of money. Suppose the input is X, which means that I bet X dollars each for the top10 riders predicted. The input of this study is preset to 100 dollars, and the final output is profit Y_1 dollars, and the profit Y_2 dollars generated by adding the trust reference will also be output.

From the distribution curve and variance of 17 features, we found that the variance of some features is too large, which will cause the model to fail to converge. Therefore, it is necessary to find a distribution that can quickly converge the model and the not too different from the original data. In Section 3, we normalized the data. The reason is to improve the convergence speed of the model and make each feature contribute to our results to the same extent. In this section, we show the results produced by using 2 different normalization methods, which are mapped to Gaussian distribution and Uniform distribution, respectively. Table 1 is the comparison of Fl scores of different training data using different feature mapping methods in different models. We can clearly find that the result of mapping to Gaussian distribution is better than Uniform distribution.

Table 1 Comparison of Fl scores of different training data using different methods (LR = Logistic Regression, NN = Neural Network, DL = Deep Learning)

ML Method		LR		NN		DL	
normalization		Gaussian	Uniform	Gaussian	Uniform	Gaussian	Uniform
Data set	Train_1	0.67	0.29	0.80	0.64	0.64	0.35
	Train_2	0.63	0.29	0.61	0.55	0.56	0.50
	Train_3	0.63	0.15	0.67	0.52	0.69	0.50
	Train_4	0.47	0.15	0.60	0.64	0.60	0.13
	Train_5	0.55	0.55	0.64	0.64	0.59	0.50

Table 2 shows the comparison of the precision and the F1 score of different training data and models. In terms of models, it can be clearly seen that the precision of the data trained by NN is higher, but the reason for the poor performance in deep learning may be that there are not enough training data, so the effect is relatively poor. Therefore, we can't blindly use deep learning for any problem. From the data point of view.

Table 2 Comparison of Precision and F1 Score Across Training Data and Models
(LR = Logistic Regression. NN = Neural Network. DL = Deep Learning.)

		Train_1	Train_2	Train_3	Train_4	Train_5
Precision	LR	1.00	0.86	0.86	0.80	0.60
	NN	1.00	0.64	0.67	0.75	0.62
	DL	0.70	0.54	0.59	0.75	0.53
F1 Score	LR	0.67	0.63	0.63	0.47	0.55
	NN	0.80	0.61	0.67	0.60	0.64
	DL	0.64	0.56	0.69	0.60	0.59

When we compare Train_1 and Train_2, we can see that the data of D2 is missing, which causes the performance of Train_2 to drop, so it shows that D2 is a very important feature. Then we observe Train_1 and Train_3, and we can see that $D3.sprint$ is missing, which causes the performance of Train_3 to drop, so it means that D3 must include $D3.climb$ and $D3.sprint$ to achieve the best performance. In the comparison of Train_1 and Train_4, you can also see the same results. We can also notice that in the comparison of Train_3 and Train_4, the importance of $D3.sprint$ is higher than $D3.climb$. Then, in the comparison between Train_2 and Train_5, it can be seen that Train_2 lacks D2, but the training results are better, indicating that adding 2 D2 at a time has a negative result on the accuracy rate. Continue to observe Train_1 and Train_2, you can see that $D2.Dau$ is missing, which causes the performance of Train_2 to drop, which shows that $D2.Dau$ is beneficial to the whole training. However, when comparing Train_1 and Train_5, it can be found that the addition of $D2.Sui$ to Train_5 causes a significant drop in precision, which also shows that only the addition of $D2.Dau$ will have a better impact on the results. According to the above, it can be seen that the importance of $D2.Dau$ is far greater than that of $D2.Sui$, $D3.sprint$ are higher than $D3.climb$, and the importance of D3 is also greater than that of D2. Therefore, we can get a ranking of importance in D2 and D3, the ranking is as follows:

$$D3.sprint > D3.climb > D2.Dau > D2.Sui \tag{14}$$

The reason for this importance may be that in the multi-day Tour *de* France, almost every day has a sprint point and a climbing point, and there will be at least one day of time trials. Time trials are good for sprinters. Therefore, the overall impact of sprinting on this multi-day schedule will be slightly greater, but there are also a lot of climbing points, so if we want the prediction to be better, it is best to take both. In addition, D2.Dau is not as important as D3, but it is also obvious that after adding training data, the precision has increased significantly. Amount optimization results of different

weights for different training data in NN are shown in Table 3, and the stake is 100 dollars.

Table 3 Amount optimization of different trust weights for different training data in NN

Methods:NN	before optimization	W = 0.8	W = 0.9	W = 1	W = 1.1	W = 1.2
Train_1	1161	1296	1296	1296	1608	1608
Train_2	-72	67	67	-18	-18	-103
Train_3	1090	1115	1115	1115	1115	1115
Train_4	534	751	751	751	751	751
Train_5	997	747	959	1042	1042	1042

Table 4 shows the profit growth rate of different weights for different training data in NN. We can find that the higher the weight of most data, the amount will increase or remain unchanged, except for Train_2, that is, when the original profit is negative, the higher the weight, the negative growth of the amount. Looking at the profit margin, the whole trend can find that as the weight increases, the best profit growth margin is no change. At the worst profit growth rate, with the increase of the weight, there is a gradual improvement, and it will not decrease until the weight is 1.2. Then, after observing the mean and the number of variance and combining the above, we select the result that the variation is the smallest and the average profit margin is not too small is used as the final weight, that is, the final trust weight of our research is 1.1. Choosing W = 1.1 can keep our profits growing positively, and the average profit margin can also maintain a certain level, and it is also the smallest under the worst profit growth margin (Worst_case).

Table 4 Various statistics of profit growth rate of different trust weights for different training data in NN

Methods:NN	W = 0.8	W = 0.9	W = 1	W = 1.1	W = 1.2
Train_1	0.11	0.11	0.11	0.38	0.38
Train_2	1.93	1.93	0.75	0.75	-0.43
Train_3	0.02	0.02	0.02	0.02	0.02
Train_4	0.40	0.40	0.40	0.40	0.40
Train_5	-0.25	-0.03	0.04	0.04	0.04
Best_case	0.40	0.40	0.40	0.40	0.40
Worst_case	-0.25	-0.03	0.02	0.02	-0.43
Mean	0.442	0.486	0.264	0.318	0.082
Variance	0.745	0.679	0.097	0.090	0.114

Table 5 shows the comparison of the final output amount of different training data (Train_1, Train_2, Train_3, Train_4 and Train_5) in NN, and the stake is also 100 dollars. It can be seen from the above results that after adding the consideration of trust, we adjusted our stake according to the trust and odds, and the profit increased, so that the amount earned by the user increased and the amount lost decreased.

Table 5 Comparison of the final output amount of different training data in NN

Methods:NN	Amount without trust	Amount with trust	Profit growth	Trust
Train_1	1161	1608	0.38	0.384
Train_2	-72	-18	0.75	0.427
Train_3	1090	1115	0.02	0.390
Train_4	534	751	0.40	0.406
Train_5	997	1042	0.04	0.343

The trust algorithm proposed in our experiment is heuristic. After the comparison in the previous section, it can be found that the use of trust optimization can increase profits. For this reason, we also present an experiment with the ideal value of trust, and we sample some data for experiments, so that we can compare the gap between our formula design and the ideal value. The sampling ratio is 40% of the original data. The results are shown in Table 6. The following results take NN as an example, and we can see the gap between our sampling trust and the ideal value is between 4% and 8%.

Table 6 Taking NN as an example, the gap between different training data and the ideal value of trust

Methods:NN	Ideal value	Trust after sampling	Gap percentage
Train_1	0.384	0.362	6
Train_2	0.427	0.405	5
Train_3	0.390	0.372	4
Train_4	0.406	0.379	7
Train_5	0.343	0.317	8

After proposing the algorithm of trust score, we re-examined the part of deep learning. In the aforementioned results, we know that the result of deep learning is the worst. We believe that the main reason is the insufficient amount of data. For this reason, we conduct an additional experiment to discuss the relationship between the amount of data and the degree of trust. We use different data sizes for training, using deep learning

as a model, and the data size adjustment method is SMOTE. We increase the data of a small number of samples to different degrees to achieve the purpose of different data sizes. Table 7 shows the comparison of the amount of different training data of on the precision and trust (the left side of the brackets of the data volume ratio is the total number of data with a label equal to zero, that is, the riders who are not in the top10, and vice versa are the riders in the top10).

Table 7 Comparison of the amount of training data on the precision and trust

	data ratio(394,36)		data ratio(394,216)		data ratio(394,394)	
	Precision	Trust	Precision	Trust	Precision	Trust
Train_1	0.26	0.3764	0.56	0.3844	0.60	0.3844
Train_2	0.38	0.4314	0.50	0.4336	0.40	0.4282
Train_3	0.50	0.3107	0.44	0.3112	0.53	0.3125
Train_4	0.38	0.4073	0.38	0.4001	0.55	0.4091
Train_5	0.46	0.3459	0.53	0.3461	0.53	0.3435

In the part of data proportion, in Train_1 and Train_2, the amount of data with label 1 is adjusted to be the same as the amount of data with label 0, that is, there are 394 results. In Train_3, Train_4 and Train_5, in the data In the process of adding the data volume to 394, the result is continuous deterioration, so we choose to reduce the increase in the data volume, so that the change is easy to see. In Train_1, it can be seen that the amount of data increases, which can increase the precision slightly, but the precision of 216 volume and 394 volume is similar, and even 216 volume is better. In the trust part, as the amount of data increases, the trust has slight increase, but it is not very obvious, and it can also be found that the trust increases or decreases with the precision rate. In Train_2, it can be seen that with the increase of the amount of data, the precision first increases, and then decreases. The increase of the amount of data can indeed increase the precision, but there is a limit, and it will decrease after the amount of data. In the trust part, with the amount of data increases, trust increases slightly, but there is also a limit, and it can also be found that trust increases or decreases with the accuracy rate. In Train_3, it can be seen that the amount of data increases, and the precision first drops a lot, and then increases again. In the part of the trust, as the amount of data increases, the trust increases slightly. In Train_4, it can be seen that the amount of data increases, and the precision first drops a lot, and then rises. In the trust part, as the amount of data increases, the trust first drops a lot, and then rises again. In Train_5, it can be seen that the amount of data increases, and the precision increases. In the trust part, as the amount of data increases, the trust first drops a lot, and then rises again.

On the whole, increasing the amount of data for a few tags can indeed increase the precision, but there is also a limit, after which it will decrease. In the part of trust, in most cases, with the increase of the amount of data, the trust increases, but the increase is not large. There is also a strong relationship between precision and trust, and in most cases, trust increases or decreases with precision. Therefore, it can be concluded that the amount of training data does affect the precision, which can be learned by observing the trust, and we may be able to use the trust as a benchmark for data augmentation research in the future.

This section will show how we actually use the cycling gambling system, the stake is 100 dollars, and we predict the results of Tour *de* France from 2019 to 2021. The final results are shown in Table 8, we can find that precision of top1 in 2019 and 2021 is about 60%, and the profit has grown after optimization, but the forecast result in 2020 is slightly less than ideal. Therefore, we speculate the result in 2020, we found that because of the epidemic, many cycling races have been suspended, and the predicted data is about 25% less than other data on average. The lack of data will also affect the training effect of NN, so the prediction results are not too good, but even so, we consider the amount after trust optimization, there is still an increase.

Table 8 Results of Tour *de* France from 2019 to 2021

	Precision	Amount without trust	Amount with trust	Profit growth	Trust
Predict_2019	0.62	974	1030	0.05	0.252
Predict_2020	0.38	529	563	0.06	0.284
Predict_2021	0.57	628	757	0.20	0.250

This section explores what problems the system will encounter and how to solve it when large amounts of data are added in the future. In the trust module of this system, KNN is used as the core. However, when KNN encounters a large amount of data, there will be a time complexity problem, so a method must be found to speed up. We refer to the methods proposed by (Johnson, 2019) to speed up our KNN, which is to implement KNN with Facebook AI Similarity Search (FAISS). According to the conclusion obtained by the combination of training data used by (Adamczyk, 2020), the training time can be reduced by 300 times, and the prediction time can be reduced by a maximum of 17 times.The final time comparison results after optimization are shown in Table 9 and Table 10. It can be found that the training time is reduced, but the prediction time is not reduced. We speculate that the main reason is the number of data, the amount of data

used is large, the model training and prediction time is longer than our research, and it is easier to see the effect of acceleration if the time is long (we can know by comparing the time complexity, assuming n data, time complexity of KNN is O(n), and the time complexity of FAISS is O(log n)). However, because of the small amount of data, the measured time is shorter, and the acceleration effect is not so obvious. Especially in the part of the prediction time, so there will be fluctuations.

Table 9 Comparison of training time using KNN and FAISS

	KNN Train time(s)	FAISS Train time(s)	Reduction factor
Predict_2019	0.000473	0.000185	2.36
Predict_2020	0.000514	0.000216	2.37
Predict_2021	0.000520	0.000166	3.13

Table 10 Comparison of predicting time using KNN and FAISS

	KNN Predict time(s)	FAISS Predict time(s)	Reduction factor
Predict_2019	0.004173	0.006644	-1.59
Predict_2020	0.005362	0.006251	-1.16
Predict_2021	0.004990	0.005860	-1.17

The above shows and analyzes the results of the entire Tour *de* France event prediction and betting reward prediction system, which can be roughly divided into two parts for discussion, the ML module and the trust module. In the ML module, NN has the best learning results, and the logistic regression is slightly inferior to NN. However, the effect of deep learning is not very good, probably because the training data is not large enough. In the trust module, we choose the ones with better performance in the ML module to conduct more in-depth research. It can be seen that in the above results, the addition of trust will indeed affect the profit.

On the other hand, in the comparison of different training data (Train_1, Train_2, Train_3, Train_4 and Train_5), Train_1 performed the best, that is, the basic data of the first type of players and $D2.Dau$, as well as the third category of rider specializations ($D3.$sprint and $D3.climb$), these factors have a greater impact on the prediction.

We also discussed some applications related to trust. We obtained the gap between our formula and the ideal value. We also discussed the relationship between the amount of training data and trust. In most cases, we can know that the accuracy and trust will increase with the amount of data, but not in a few cases. The reason for the above may be that the data we increase by using SMOTE may not be very good, because it is not in the

same way as old data was obtained. Based on the above results, and watching how our training data increases, the training results are still inferior to NN, so in betting systems, we still recommend using NN as our model. We also show our complete gambling system based on the above conclusions, and predict the results of Tour *de* France from 2019 to 2021, and also show that through our gambling system, the profit can indeed be optimized.

5. Conclusion

AI has changed people's lives, and many studies have been generated. This research focuses on cycling races and analyzes the trust issues generated by current AI. A gambling prediction system is constructed as a whole. In terms of predicting Tour *de* France, we succeeded in predicting whether each rider was in the top10 and using different training profile to achieve multi-day event predictions. In terms of ML modules, we try different ML methods, and finally NN has the best results, and finds the combination of training data that can make the model result the best. In terms of trust, we propose a new trust algorithm. On the whole, we first get a trust score, which allows users to evaluate the entire model, and we apply this score to the part that maximizes profit, and the results achieve what we want.

Reference

Adamczyk, J. (2020). *Make kNN 300 times faster than Scikit-learn's in 20 lines!* https://towardsdatascience.com/make-knn-300-times-faster-than-scikit-learns-in-20-lines-5e29d74e76bb

De Spiegeleer, E. (2019). *Predicting cycling results using machine learning.*

Das, S. (2019). *Trust issues: Is AI black box creating a black future?* https://analyticsindiamag.com/trust-issues-is-ai-black-box-creating-a-black-future/

David, J. A., Pasteur, R. D., Ahmad, M. S., & Janning, M. C. (2011). NFL prediction using committees of artificial neural networks. *Journal of Quantitative Analysis in Sports, 7*(2).

Gilchrist, S. (2020). *A brief history of cycling and cycling betting.* https://kayokokishimoto.com/cycling-betting/

Gabriele, M. C. (2011). *The golden age of bicycle racing in New Jersey.* The History Press.

Hoff, K. A., & Bashir, M. J. (2015). Trust in automation: Integrating empirical evidence on factors that influence trust. *Human Factors, 57*(3), 407–434.

Johnson, J., Douze, M., & Jégou, H. (2019). Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data, 7*(3), 535–547.

Kataoka, Y., & Gray, P. (2018). Real-time power performance prediction in Tour de France. In *International Workshop on Machine Learning and Data Mining for Sports Analytics* (pp. 121–130). Springer.
Kholkine, L., De Schepper, T., Verdonck, T., & Latré, S. (2020). A machine learning approach for road cycling race performance prediction. In *International Workshop on Machine Learning and Data Mining for Sports Analytics* (pp. 103–112). Springer.
Lee, E. J. (2021). How do we build trust in machine learning models?
Logg, J., Minson, J., & Moore, D. A. (2018). Do people trust algorithms more than companies realize? *Harvard Business Review, 26*.
Mutijarsa, K., Ichwan, M., & Utami, D. B. (2016). Heart rate prediction based on cycling cadence using feedforward neural network. In *2016 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)* (pp. 72–76). IEEE.
Samek, W., Wiegand, T., & Müller, K.-R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models.
Sheehan, M. (2018). *Tour de Suisse vs. Criterium du Dauphiné: What's the best tour prep?* https://www.flobikes.com/articles/6206572-tour-de-suisse-vs-criterium-du-dauphine-whats-the-best-tour-prep
Soneja, A. (2019). *Artificial intelligence in sports: A smarter path to victory.* https://www.cio.com/article/3400877/artificial-intelligence-in-sports-a-smarter-path-to-victory.html
Thabtah, F., Zhang, L., & Abdelhamid, N. (2019). NBA game result prediction using feature analysis and machine learning. *Annals of Data Science, 6*(1), 103–116.
Thiebes, S., Lins, S., & Sunyaev, A. (2021). Trustworthy artificial intelligence. *Electronic Markets, 31*(2), 447–464.
Valero, C. S. (2016). Predicting win-loss outcomes in MLB regular season games — A comparative study using data mining methods. *International Journal of Computer Science in Sport, 15*(2), 91–112.