Problem Chosen	2025 MCM/ICM Summary Sheet	Team Control Number
C		2524908

Summary

Abstract

In this paper, we construct several models to predict the success of the 2028 Los Angeles Olympic Games. Our dataset contains the total scores for nearly 1000 countries and regions from 1896 to 2024. We built an artificial neural network model and a multiple linear regression model to predict the results of the 2028 Olympic Games. Then, we used the entropy-weighted method to integrate these two different outcomes to gain the synthetic results. We also find the relationship between the events and the number of medals gained and compare the performance of each country in the 2024 Olympic Games and their performance in the 2028 Olympic Games to determine which country will perform better and which country will perform worse. Finally, we found the great coach effect in the past Olympic Games, which brought the participating countries more medals.

Keywords: Multiple Linear Regression; Neural Network; Entropy-weighted Method

Contents

1 Introduction

1.1 Background

The Olympic Games have long served as a global stage where athletes from diverse backgrounds come together to compete at the highest level. Scheduled to take place in Los Angeles, California, the 2028 Summer Olympics mark the city's third time hosting this prestigious event, following its successful iterations in 1932 and 1984.

As the Games approach, attention is increasingly focused on the potential medal outcomes, which reflect not only the athletes' dedication but also the geopolitical, cultural, and economic forces shaping global sports. Nations are investing heavily in sports programs, science-backed training methods, and athlete development, with the aim of achieving their position on the medal table. At the same time, emerging powers in international sports are challenging the dominance of traditional medal-winning nations, hinting at a potential reshaping of the competitive landscape in 2028.

This paper explores these dynamics within the unique context of the Los Angeles Games, including its innovative event lineup, the past performances of each country and the composition of each national team.

1.2 Framework of Analysis

In this paper, we established multiple models to predict the medals earned of each country in the 2028 Olympic Games held in Los Angeles. The first model we take is the artificial neural network (ANN). The neural network program is constructed using Python and trained with the databases provided by the organizer of this competition. The databases we used included data about the names of athletes and the countries they represent, the hosting country, the medals awarded, and the hosting year from 1896 to 2024 of each country. The other model we take is multiple linear regression (MLR). In this model, we find the linear relationships between the variables in the database and the medals earned of each country. To find the synthetic results from these two outcomes of different methods, we take the entropy-weight method (EWM) to calculate the weight between the data from each method. Eventually, we will take the weighted average value to obtain the final result.

1.3 Comments on Data Provided

There have been many inconsistencies we observed in the provided data, many of which were not addressed in the problem statement. Nonetheless, we have found ways to circumvent these errors to create working models regardless. First, we must address the issue of the 1906 Summer Olympics. While running tests on the multidimensional regression model we found an error in which the model handled one specific date, namely the 1906 Olympics. After closer reading, it came to out attention that the 1906 Olympics did not follow the standard 4 year gap between events. This was cumbersome because this information was only stored in the athletes data file, and not mentioned in neither the medal count file nor the host file. This deleted crucial information on which the regression model bases its predictions on. We have elected not to include those Olympics in the re-

gression model, due to the stipulation not to use any external data (i.e. data from sources others than the ones provided). It had been included, however, in the neural network model, as there need not be for extremely precise definitions compared to the regression model; however, this might be contributing to some amount of error (formally called *loss* in machine learning environments). We have also found errors in the files themselves. There are certain characters at the end of some team names that inhibit complete data analysis on the unmodified provided files. These include, but are not limited to, the characters "¬†" being concatenated to the end of country names in the `medal_counts.csv` file for the 1932 and 1960 Olympics, the existence of "subteams" like Japan-1 or United States-13, or the aforementioned lack of host data for the special 1906 Summer Olympics. Although it is difficult to predict what influence these errors had on the neural network model, we found ways to circumvent these issues when working with the regression model.

2 Analysis of the Problem

2.1 The Artificial Neural Network

The artificial neural network was coded in Python 3.12.0, with the help of standard machine learning (ML) and data analysis libraries: `Pandas`, `NumPy`, `scikit-learn` and `TensorFlow`. The raw code may be found in Appendix A, or by clicking here. The Artificial Neural Network (henceforth abbreviated as ANN) was trained on three of the four provided files: `summerOly_atheletes.csv`, `summerOly_hosts.csv`, `summerOly_medal_counts.csv`. The ANN was designed with three layers, the input layer (with 32 neurons), the middle layer (with 16 neurons) and the output layer (with 1 neuron). After each training iteration, the ANN prunes (omits) random neurons and connections (20% for this ANN) to prevent overfitting and force the ANN to learn the patterns visible in the training data. This is done to prevent the ANN from depending on any specific neuron, which is common in smaller ANN's (like the one created for this paper) or with a relatively small amount of training data (also the case in this problem). An example of a pruned neural network is shown in Figure 1. The data followed the standard 90/10 split, whereby 90% of the data was used to train the model, and 10% to test the models accuracy. Rectified Linear Unit (ReLU) was chosen as the activation function for the first 2 ANN layers, due to the many benefits it provides over other functions like Sigmoid, which tend to slow the model down during backpropagation, since it confines all inputs into the range $[0, 1]$. For very large or small input values, the gradient is disproportionally closer to 0, which significantly slows down the updating of weights between neurons. Although many slightly different variations of the ReLU function exist, the ANN used the function defined as $x \mapsto \max(0, x)$.

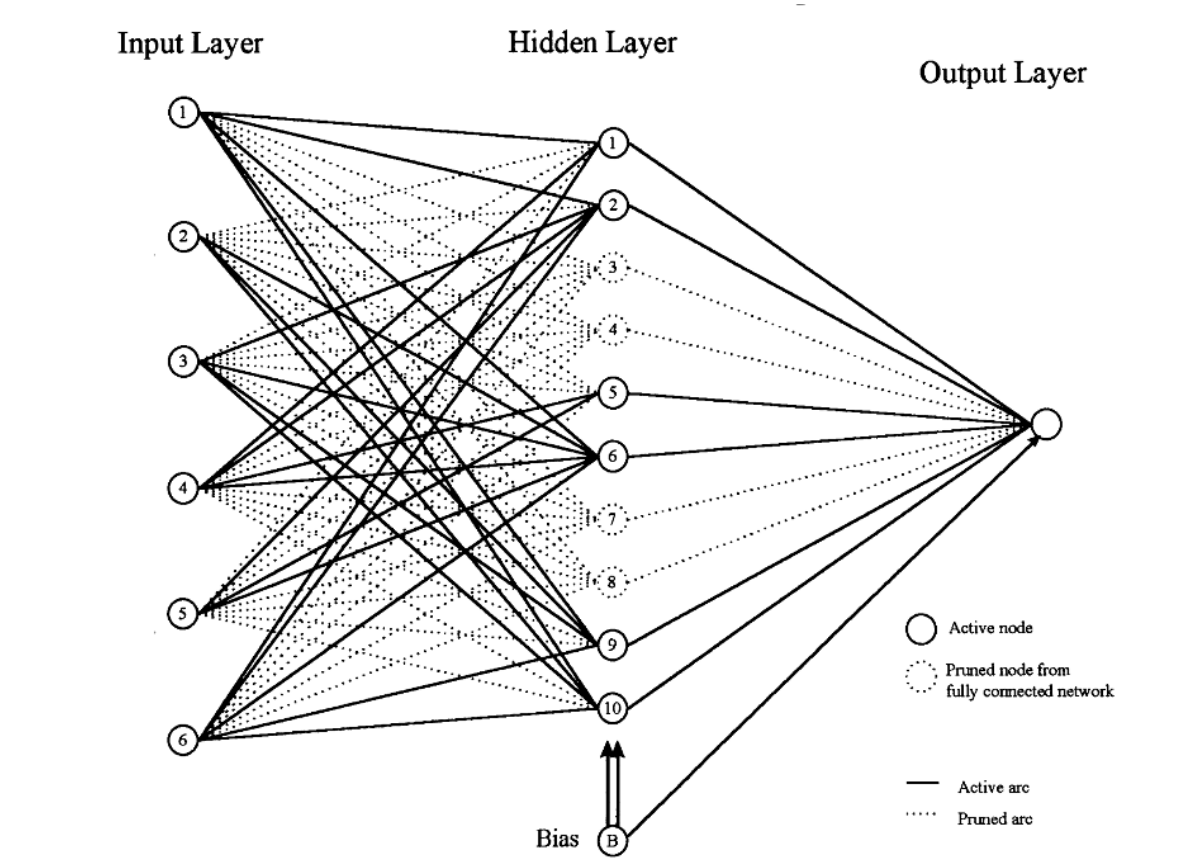

Figure 1: Example of pruned neural network

The final layer was standardized using the Softmax function, again as is custom in multilayer classification neural networks. Softmax standardizes the outputs to be a valid probability distribution, defined as:

$$\text{Softmax}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{1}$$

The ANN also used three additional functions to calculate its effectiveness. The optimizer Adam (Adaptive Moment Estimation) function was used to minimize the loss dynamically by changing the weights and biases of neurons and connections. It ensures that the neural network converges to a reasonable solution efficiently. It is defined mathematically below:

$$\begin{aligned} \omega_{t+1} &= \omega_t - \alpha m_t \\ m_t &= \beta m_{t+1} + (1-\beta)\left[\frac{\partial L}{\partial \omega_t}\right] \end{aligned} \tag{2}$$

Table 1: Variable Definitions for Equation (2)

Variable	Definition
t_0	Initial time, $t_0 = 0$
m_t	Aggregate of gradients at t ($m_{t_0} = 0$)
m_{t-1}	Aggregate of gradients at t_{previous}
ω_t	Weights at t
ω_{t+1}	Weights at $t \equiv t + 1$
α_t	Learning rate at t
∂L	Derivative of the loss function
$\partial \omega_t$	Derivative of weights at t
β	Moving average parameter, const.

The loss function was chosen to be *Binary Cross Entropy* (also known as *log loss*). The neural network was punished when loss was high, which leads to the ANN learning the data. The Binary Cross Entropy function is defined as:

$$\text{BCE} = -\frac{1}{N}\sum_{i=0}^{N}\left[y_i \log(p_i) + (1 - y_i)\log(1 - p_i)\right] \tag{3}$$

Table 2: Variable definition for Equation (3)

Variable	Definition
N	Number of instances
y_i	True label for instance i
p_i	Model predicted probability of instance i

2.1.1 Model Performance

Full results of ANN predictions may be found in Section 3. This section will be dedicated to providing the results for the ANN itself. The ANN used $n = 50$ epochs for training to ensure the maximum accuracy of the model and minimize the risk of potential overfitting. The final values for key metrics are shown in the table below.

Table 3: Key ANN metrics

Metric	Value
Training Accuracy	0.9976
Training Loss	0.0173
Validation Accuracy	0.9250
Validation Loss	0.2187

Although it is impossible to fully understand how the neural network learned the data, the above metrics give insight into its workings on training and validation (testing) data. Looking at the training metrics first, it becomes clear the model handles seen data exceptionally well, with an extremely high training accuracy and very low training loss. Usually, this means the neural network is extremely well-trained. In this case, however, it may point towards overfitting since the winners of events are (somewhat) random. Having a 0.9976 and 0.0173 training accuracy and loss on a slightly random data set is the first indication of overfitting in the model. However, the ANN still fairs well with data it has not yet seen, as evidenced by the relatively high validation accuracy. The increase in loss value from training to validation does support the overfitting hypothesis. However, it is well within accepted ranges for well-trained neural networks, meaning the predictions of the ANN may be accepted as valid.

2.2 Multiple Linear Regression

Since ANNs are designed for non-linear data and trends (due to non-linear activation functions), we thought it best to also model the given data in a way that could capture any linearity in the structure of our given data or within any trends in the data. Previous work that we on using linear regression to make predictions about the Olympics made use of data that was not available in our dataset, such as socioeconomic metrics like as GDP and HDI (Halsey 2009), and details on athletes like height and weight (Moolchandani et al. 2024). Since we did not have access to this kind of data in this problem, we instead tested for correlations between each country's medal counts (bronze, silver, gold, and total) and the following parameters:

`Athletes`
: Number of athletes that competed for the country.

`Sports`
: Number of sports that the country participated in.

`Events`
: Number of events that the country participated in.

`Is.host`
: A flag that is 1 if the country is the host, and 0 otherwise.

`Mean.Sex`
: For each country's athletes in a given year, we set their sex to 0 if they were female and 1 if they were male. `Mean.Sex` gives the mean of these values. Thus, the higher this predictor, the larger the percentage of athletes was male.

`Mean.Sport.Participation`
: For each year, we made a dictionary that counted the number of countries that competed in given sports. For each country in a given year, we indexed this dictionary only at the sports that this country participated in, and took the mean of the corresponding values. Intuitively, this could be thought of as the "mean popularity" of the sports that this country participated in for this year.

`Mean.Event.Participation`
: Same as `Mean.Sport.Participation`, only with events.

`Bronze.Last.Olympics`
: Number of Bronze medals won in the last year.

`Silver.Last.Olympics`
: Number of Silver medals won in the last year.

`Gold.Last.Olympics`
: Number of Gold medals won in the last year.

The full table containing all of these parameters for each country in each year can be found in `processed_data.csv`, and the code that generated this table can be found in `process_data.R`. The full model also contains a few interaction terms between these predictors, which will be denoted by the predictors' names concatenated with a colon ":" separator. As a final note with respect to the data, we found 20 outliers in the data that are "extreme" with respect to the dependent variable according to studentized deleted residuals, and none that are "extreme" with respect to any independent variables according to the hat values metric.

Using these parameters and their respective data, we constructed four multiple linear regression models, one for each medal type and another for total medals[1]:

Table 4: Full MLR Model

	Dependent variable:			
	Bronze	Silver	Gold	Total
	(1)	(2)	(3)	(4)
Constant	3.850***	3.875***	3.681***	11.406***
	p = 0.0001	p = 0.0001	p = 0.002	p = 0.00002
Athletes	0.034***	0.042***	0.039***	0.115***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Sports	−0.154***	−0.121***	−0.109***	−0.384***
	p = 0.00002	p = 0.001	p = 0.008	p = 0.0001
Events	−0.003	−0.034***	−0.038***	−0.075***
	p = 0.684	p = 0.00002	p = 0.00002	p = 0.0004
Is.host	21.610***	21.297***	17.014***	59.920***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000

[1]All tables in section 2.2 were generated by the R package `stargazer`—see Hlavac 2022 for a preview.

Mean.Sex	-2.654^{***}	-2.759^{***}	-2.827^{***}	-8.240^{***}
	p = 0.002	p = 0.001	p = 0.003	p = 0.0003
Mean.Sport.Participation	-0.001^{***}	-0.001^{**}	-0.001^{*}	-0.002^{**}
	p = 0.006	p = 0.047	p = 0.096	p = 0.015
Mean.Event.Participation	-0.002	-0.003^{**}	-0.002	-0.006^{**}
	p = 0.117	p = 0.015	p = 0.130	p = 0.037
Bronze.Last.Olympics	0.182^{***}	0.127^{***}	0.018	0.328^{***}
	p = 0.00001	p = 0.002	p = 0.693	p = 0.003
Silver.Last.Olympics	-0.015	0.003	-0.001	-0.013
	p = 0.739	p = 0.941	p = 0.988	p = 0.920
Gold.Last.Olympics	0.277^{***}	0.365^{***}	0.627^{***}	1.270^{***}
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Events:Is.host	-0.112^{***}	-0.095^{***}	-0.046^{***}	-0.252^{***}
	p = 0.000	p = 0.000	p = 0.004	p = 0.000
Is.host:Mean.Event.Participation	0.003	-0.001	-0.009	-0.006
	p = 0.599	p = 0.862	p = 0.246	p = 0.718
Observations	1,045	1,045	1,045	1,045
R^2	0.756	0.772	0.776	0.812
Adjusted R^2	0.754	0.769	0.774	0.810
Akaike Inf. Crit.	5,634.121	5,623.582	5,939.055	7,747.321
Bayesian Inf. Crit.	5,703.446	5,692.907	6,008.380	7,816.646
Residual Std. Error (df = 1032)	3.560	3.542	4.119	9.784
F Statistic (df = 12; 1032)	266.964^{***}	290.373^{***}	298.126^{***}	371.542^{***}
Note:				*p<0.1; **p<0.05; ***p<0.01

We see that nearly all of our predictors have very small p-values, meaning nearly all of them are significant, including the metrics related to sports and events. The only exception is `Silver.Last.Olympics`, which shows the opposite effect—its p-values are very large, indicating that the number of silver medals a country has is. When analyzing our correlations, we focus only on the model with the total number of medals as the dependent variable to reduce redundancy, as our analyses were identical for all models.

2.2.1 Backwards Model Selection

This model has a great deal of unnecessary predictors, as shown by the low p-values. We will use the backwards selection algorithm provided by the R package `MASS` to select

smaller models. After backwards selection, we have the following information of our models:

Table 5: Reduced models after backwards selection

	Dependent variable:			
	Bronze	Silver	Gold	Total
	(1)	(2)	(3)	(4)
Athletes	0.034***	0.042***	0.039***	0.114***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Sports	−0.152***	−0.122***	−0.115***	−0.388***
	p = 0.00002	p = 0.001	p = 0.005	p = 0.0001
Events	−0.004	−0.034***	−0.036***	−0.074***
	p = 0.646	p = 0.00002	p = 0.00003	p = 0.0005
Is.host	21.661***	21.278***	16.811***	59.787***
	p = 0.000	p = 0.000	p = 0.00000	p = 0.000
Mean.Sex	−2.630***	−2.766***	−2.866***	−8.264***
	p = 0.002	p = 0.001	p = 0.003	p = 0.0003
Mean.Sport.Participation	−0.001***	−0.001**	−0.001*	−0.002**
	p = 0.006	p = 0.045	p = 0.082	p = 0.014
Mean.Event.Participation	−0.002	−0.003**	−0.002	−0.006**
	p = 0.126	p = 0.014	p = 0.111	p = 0.034
Bronze.Last.Olympics	0.177***	0.128***		0.321***
	p = 0.00000	p = 0.0003		p = 0.002
Gold.Last.Olympics	0.270***	0.367***	0.638***	1.265***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Events:Is.host	−0.110***	−0.095***	−0.050***	−0.255***
	p = 0.000	p = 0.000	p = 0.002	p = 0.000
Constant	3.803***	3.889***	3.770***	11.458***
	p = 0.0001	p = 0.0001	p = 0.001	p = 0.00002
Observations	1,045	1,045	1,045	1,045
R^2	0.756	0.771	0.776	0.812
Adjusted R^2	0.754	0.769	0.774	0.810

Akaike Inf. Crit.	5,630.521	5,619.619	5,934.577	7,743.463
Bayesian Inf. Crit.	5,689.943	5,679.041	5,989.046	7,802.884
Note:				*p<0.1; **p<0.05; ***p<0.01

We see that the predictors `Silver.Last.Olympics` and `Is.host:Mean.Event.Participation`, the interaction term, were dropped from every model, and `Bronze.Last.Olympics` was dropped specifically from the Gold model. We will use these models in all predictions and in the following sections.

2.2.2 Multicollinearity

Measuring multicollinearity in each predictor is important to determine how independent their effects are on the full model. If a given predictor exhibits multicollinear, it is common for it to have a small p-value in a MLR model, but a small p-value in a SLR model, or vice versa, thereby being inconsistent. We measure multicollinearity using Variance Inflation Factors (VIF). The functionality to calculate this is provided by the R package `cars`. The VIFs for each predictor from each model is given below.

Table 6:

	VIFs			
Predictor	Bronze	Silver	Gold	Total
Athletes	13.824	13.824	13.823	13.824
Sports	6.685	6.685	6.671	6.685
Events	15.513	15.513	15.054	15.513
Is.host	13.095	13.095	13.081	13.095
Mean.Sex	1.797	1.797	1.797	1.797
Mean.Sport.Participation	1.439	1.439	1.437	1.439
Mean.Event.Participation	1.346	1.346	1.346	1.346
Bronze.Last.Olympics	4.909	4.909	1.844	4.909
Gold.Last.Olympics	4.428	4.428	11.618	4.428
Events:Is.host	11.628	11.628	13.823	11.628

We usually note that a predictor exhibits multicollinearity if its VIF value exceeds 10. Correspondingly, we see the value for `Athletes` exceedes 10, indicating multicollinearity in this predictor. We do see multicollinearity in `Events` and `Is.host`, but these are directly correlated to `Events:Is.host`, so this is expected. Further, we can inspect the effects of these predictors on their own with each model:

Table 7: SLR on Athletes

	Dependent variable:			
	Bronze	Silver	Gold	Total

	(1)	(2)	(3)	(4)
Athletes	0.044***	0.044***	0.049***	0.014***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Constant	−0.689***	−1.077***	−1.642***	−3.408***
	p = 0.00002	p = 0.000	p = 0.000	p = 0.000
Observations	1,419	1,419	1,419	1,419
R^2	0.602	0.563	0.500	0.5879
Adjusted R^2	0.602	0.563	0.499	0.5876
Residual Std. Error (df = 1417)	4.306	4.678	5.956	13.87
F Statistic (df = 1; 1417)	2,145.070***	1,826.512***	1,414.241***	2,021.236***
Note:		*p<0.1; **p<0.05; ***p<0.01		

Table 8: Isolated MLR on Events, Is.host, and Interaction Term

	Dependent variable:			
	Bronze	Silver	Gold	Total
	(1)	(2)	(3)	(4)
Events	0.084***	0.077***	0.084***	0.244***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Is.host	27.476***	28.005***	20.255***	75.736***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Events:Is.host	−0.124***	−0.103***	−0.043**	−0.270***
	p = 0.000	p = 0.000	p = 0.031	p = 0.00000
Constant	−1.367***	−1.392***	−1.839***	−4.598***
	p = 0.000	p = 0.000	p = 0.000	p = 0.000
Observations	1,419	1,419	1,419	1,419
R^2	0.537	0.485	0.421	0.504
Adjusted R^2	0.536	0.484	0.420	0.503
Residual Std. Error (df = 1415)	4.649	5.083	6.409	15.214
F Statistic (df = 3; 1415)	547.086***	444.323***	343.331***	480.140***
Note:				*p<0.1; **p<0.05; ***p<0.01

Thus, we see that with respect to p-values, although these predictors exhibit multicollinearity, they are relevant to their models regardless of the presence of all other predictors.

2.2.3 Error Variance Constancy (Total model)

We use a Modified Levene Test (see Appendix A for details) on each predictor in the Total model to detect non-constant error variance. This gave the following results:

Table 9:

Athletes	0.076
Sports	0.021
Events	0.019
Is.host	0.477
Mean.Sex	0.004
Mean.Sport.Participation	0.030
Mean.Event.Participation	0.041
Bronze.Last.Olympics	0
Gold.Last.Olympics	0

Using a level $\alpha = 0.05$ for this test, we see that all predictors have non-constant error variance except for `Athletes` and `Is.host`.

2.2.4 Influential points

According to DFFITS, we found 12 influential points (see `DFFITS.csv`); according to Cook's Distance, we found 4 influential points (see `cooks_distance.csv`); finally, according to DFBETAS, we found 20 influential points.

3 Results Integration

The projected medal table for the 2028 Olympics from both models is given in the table below. Note that in the predictions, the Total represents the sum of Bronze, Silver, and Gold columns, and the columns are rounded after summing. Thus, the displayed medals do not always sum to the total.

Table 10: Predicted 2028 Olympics Medal Table

Name of Country	Artificial Neural Network				Multiple Linear Regression			
	Gold	Silver	Bronze	Total	Bronze	Silver	Gold	Total
United States	73	38	33	144	29	32	40	101
Great Britain	26	23	11	60	16	15	14	44
France	29	10	21	42	22	21	21	64
Italy	26	7	4	28	14	12	12	39
Germany	15	6	6	23	14	14	14	42
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
Egypt	1	0	1	2	2	2	1	5
Tajikistan	0	0	0	0	1	1	0	2
Qatar	0	0	0	0	0	0	0	0
Botswana	0	0	0	0	0	0	0	0

Results from the MLR model with 95% confidence interval ranges can be found in `predicted_medal_table.csv`, with the fitted value ending in `.fit`, lower bounds ending in `.lwr`, and upper bounds ending in `.upr`.

The prediction of the neural network was standardized relative to the number of disciplines projected to be in the 2028 Olympics. Clearly, if more events are in a given Olympic, the higher the total number of medals will be, and therefore the total number of medals won by each team will be greater. Since the events and teams for the 2028 Olympics have not yet been published, we were forced to make assumptions regarding these metrics. For the medal predictions, the values were adjusted based on the number of events we assume will happen in 2028, which we took to be the exact same as in 2024. The model does account, however, for a differing number of disciplines, and can be parsed into the function to change to predictions.

However, due to the differences of these two algorithms, many differences exist between the values obtained. To solve the problem of differences in the data obtained from different methods, we need to use particular methods to combine these outcomes to get the results. To do this, we adapt the entropy-weighted method to calculate the synthetic result.

3.1 Entropy Weight Method Model

The entropy-weighted model is an evaluation model that determines the weights of various indicators based on the differences in the information contained in each indicator and combines the rankings derived from the proximity to the ideal solution. The specific operational flow chart is as follows:

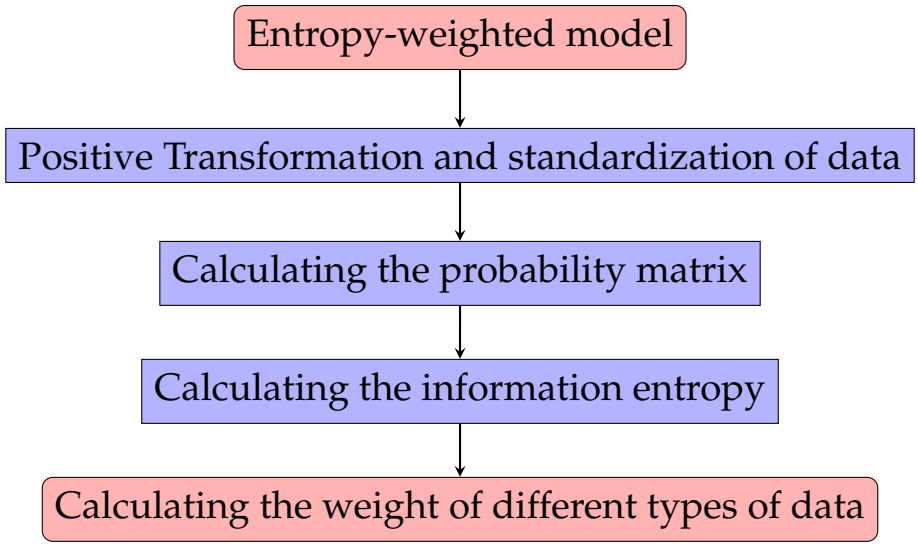

Figure 2: Flowchart of The Process of EWM

In most previous competitions, most countries did not win any medals, and only very few countries gained the most, which looks like a Gamma distribution, as the figure shown below:

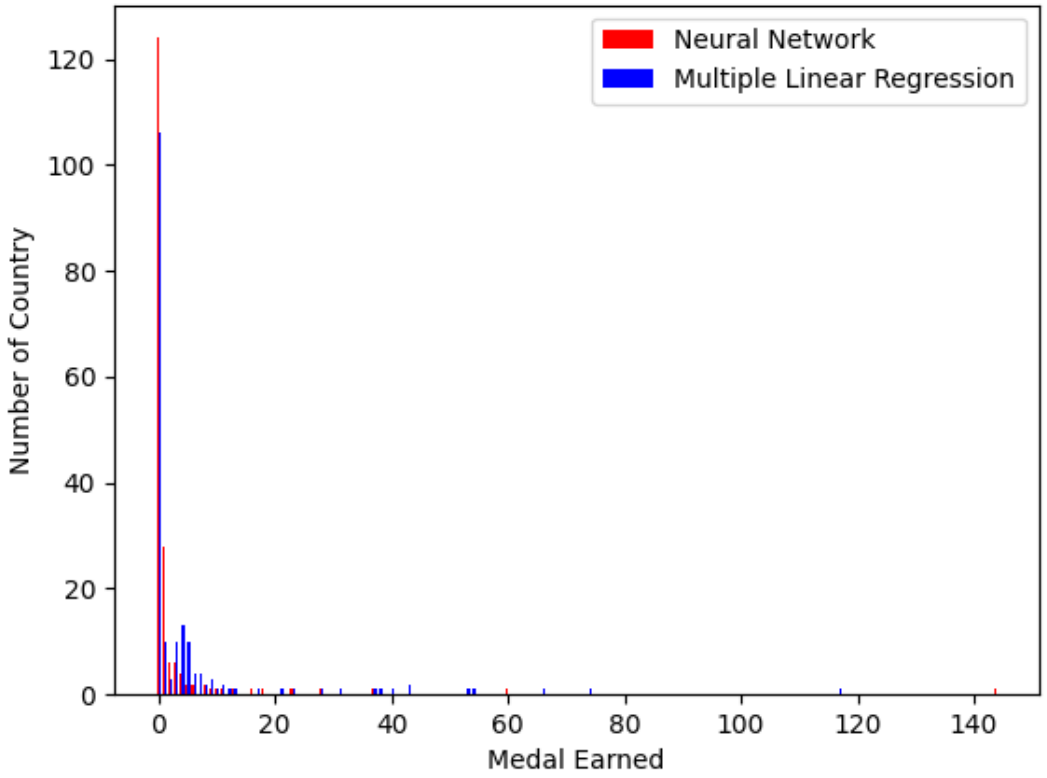

Figure 3: Distributions of Total Medals Earned from both methods

Thus, the criteria to determine which result is more likely to be correct is to see which results about the total medals gained by each country match the Gamma distribution.

$$f(x;k,\theta) = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}. \tag{4}$$

In this case, we are going to do the positive transformation by using the formula below:

$$x_i = \frac{f(x_i) - \min(f(x))}{\max(f(x)) - \min(f(x))} \tag{5}$$

The x_i in(5) is the number of total medals a particular country gains under a specific method. The data after the positive transformation is in the table below:

Table 11: Data about total medals earned after positive transformation

Name of the Country	**Artificial Neural Network**	**Multiple Linear Regression**
United States	0	0
Great Britain	2.16×10^{-7}	3.87×10^{-7}
France	4.64×10^{-7}	2.23×10^{-10}
Italy	3.17×10^{-4}	7.82×10^{-6}
German	3.02×10^{-3}	2.89×10^{-7}
$\vdots$	$\vdots$	$\vdots$
Egypt	9.99×10^{-1}	5.57×10^{-1}
Kajikkistan	$8.24 \times 10{-}1$	9.99×10^{-1}
Qatar	$8.24 \times 10{-}1$	$8.24 \times 10{-}1$
Botswana	$8.24 \times 10{-}1$	$8.24 \times 10{-}1$

After the positive transformation, we need to normalize the data by using the formula below:

$$\hat{x}_i = \frac{x_i}{\sqrt{x_1^2 + x_2^2 + \cdots + x_n^2}} \tag{6}$$

In the formula (6), n is the number of the country. Since we obtain the total medals earned by using two different method, we can use a two-dimension matrix to contain these value:

$$\tilde{x} = \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \\ \vdots & \vdots \\ x_{(n-1)1} & x_{(n-1)2} \\ x_{n1} & x_{n2} \end{bmatrix} \tag{7}$$

The first column represents the total medals earned as a prediction from the artificial neural network, and the second column represents the total medals earned from the pre-

diction of multiple linear regression. Now, our goal changes to find the matrix:

$$\tilde{z} = \begin{bmatrix} z_{11} & z_{12} \\ z_{21} & z_{22} \\ \vdots & \vdots \\ z_{(n-1)1} & z_{(n-1)2} \\ z_{n1} & z_{n2} \end{bmatrix} \tag{8}$$

where

$$z_{ij} = \frac{x_{ij}}{\sqrt{x_{1j}^2 + x_{2j}^2 + \cdots + x_{nj}^2}} \tag{9}$$

After gaining the normalized data of the total medals earned, we calculating the Weight matrix:

$$\tilde{p} = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \\ \vdots & \vdots \\ p_{(n-1)1} & p_{(n-1)2} \\ p_{n1} & p_{n2} \end{bmatrix} \tag{10}$$

where

$$\tilde{p}_{ij} = \frac{z_{ij}}{\sum_{i=1}^{n} z_{ij}} \tag{11}$$

And then, we calculate the information entropy, e_j of each method (ANN and MLR):

$$e_j = -\frac{1}{\ln n} \sum_{i=1}^{n} p_{ij} \ln(p_{ij}) \quad (j = 1, 2) \tag{12}$$

Then, we calculate the information utility value of each type of method (ANN and MLR):

$$d_j = 1 - e_j. \tag{13}$$

Finally, we get the weight of each method by normalizing the utility values:

$$w_j = \frac{d_j}{d_1 + d_2} \quad (j = 1, 2) \tag{14}$$

The entropy-weighted method was coded in Python 3.12.6 with data analysis libraries NumPy, SciPy, and Pandas. The weights of each method are shown in the table below.

Table 12: Weight of Each Method

Artificial Neural Network	**Multiple Linear Regression**
0.52277079	0.47722921

Based on the composite weights, we calculate the weighted average of the medaas earned obtained by different methods for the same country:

$$\text{Medals earned} = w_{\text{ANN}} R_{\text{ANN}} + w_{\text{MLR}} R_{\text{MLR}}. \tag{15}$$

R_{ANN} is the total medals earned by a certain country calculated by the artificial neural network, and R_{MLR} is the total medals earned by the same country calculated by the multiple linear regression. We use this to derive the final medals earned for each country as show in the next section.

4 Projections

Table 13: Projected Medals to be Earned for Each Country in the 2028 Olympics

Country	Bronze	Silver	Gold	Total
United States	33	38	60	131
Great Britain	13	19	20	52
France	22	7	24	53
Italy	5	9	19	33
German	9	10	14	33
⋮	⋮	⋮	⋮	⋮
Egypt	1	1	1	3
Tajikistan	0	1	0	1
Qatar	0	0	0	0
Botswana	0	0	0	0

4.1 Countries that are Projected to Improve

To determine which country needs to work hard to improve, we established a metric, performance score, to measure a country's performance in the Olympics. We assigned unique performance scores to gold, silver, and bronze medals. The performance score for a gold medal is 5 points, for the silver medal is 3 points, and for the bronze medal is 2 points. We find each country's total score by adding the score from these three kinds of medals together to see the final performance score of each country in this Olympic Games. We take the data in the medal table of each country in 2024 to calculate their performance score in the 2024 Olympic competition. Meanwhile, we repeat this calculation again based on the predicted data about each country's number of medals gained to get their performance score in the 2028 Olympic Games.

A decrease in a country's performance score indicates that the country needs to put in effort to improve its current sports situation. Conversely, an increase in the performance score suggests that the country's current sports condition will likely lead to better results in the next Olympic Games.

According to the comparison of performance scores, the following countries on the list about their performance in the 2028 Olympic games are projected to improve:

Table 14: Countries that are Projected to Improve in the 2028 Olympic Game

Rank	Country	Bronze	Silver	Gold	Total	Score
1	France	23	22	21	66	217
2	Australia	18	17	19	54	182
3	Japan	17	17	19	53	180
4	Germany	14	14	14	43	140
5	Ireland	14	14	12	40	130
6	Netherlands	12	12	13	37	125
7	Spain	11	11	10	31	105
8	Canada	10	9	9	28	92
9	Poland	5	4	3	12	37
10	Belgium	4	4	3	11	35
11	Denmark	4	3	3	9	32
12	Jamaica	3	3	2	8	25
13	Serbia	2	2	3	7	25
14	Kosovo	2	2	2	7	20
15	India	3	2	1	7	17
16	Chinese Taipei	3	2	2	7	22
17	North Korea	3	2	2	6	22
18	Ethiopia	2	2	2	6	20
19	Hong Kong	2	2	2	5	20
20	Bahrain	1	2	2	5	18
21	Pakistan	2	2	2	5	20
22	Ivory Coast	2	2	2	5	20
23	Indonesia	2	2	2	5	20
24	Egypt	2	1	1	5	12
25	Philippines	2	1	2	5	17
26	Kyrgyzstan	2	2	1	4	15
27	Malaysia	2	1	1	4	12
28	Slovenia	2	1	2	4	17
29	Albania	1	1	1	4	10
30	Uganda	1	1	1	4	10
31	Singapore	1	1	1	4	10
32	Dominica	1	1	1	3	10
33	Cabo Verde	1	1	1	3	10
34	Moldova	2	1	1	3	12
35	Guatemala	1	1	1	3	10
36	Panama	1	1	1	3	10
37	Mongolia	1	1	1	3	10
38	Saint Lucia	1	1	1	3	10
39	Peru	1	1	1	2	10
40	Slovakia	1	1	0	1	5

4.2 New Medalists

To find the new medalists, we analyzed the `medal_counts.csv` file, to find countries which historically have not won even a single medal in any Summer Olympics. However, after summing up the total medal counts across all year from 1896 to 2024, it turns out no countries from the provided file have exactly 0 medals. Every nation in the `medal_counts.csv` have earned at least 1 medal throughout all the given Summer Olympics, meaning no countries can possibly be new medalists. We are not sure, however, if this is by design of the data set - it is possible the competition organizer has purposefully committed countries which earned no medals in a given year. It is entirely possible that the dataset was constructed so countries with no medals won were not included in the data set. If so, a more thorough check of results of all countries throughout all Summer Olympics must be brought out; however, the competition prohibits use of external data. Therefore, basing our on the fact that no 'medalless' countries in the `medal_counts.csv` file were found, the conclusion emerges that no new medalists can be expected in 2028.

5 Model Insights

5.1 Significance of Athletes' Sex

In our MLR model, we found a strong negative linear correlation between `Mean.Sex`, that being the proportion of male athletes in a country, and all medal counts. In other words, a higher proportion of female athletes is linearly correlated with higher medal counts of all types and a higher total medal count.

5.2 Is Host Influence

Both models found that a country being the host of the Olympics in a given year makes them more likely to win more medals. Although not possible to predict fully what influence this had on the ANN, there is a relatively large disparity between predicted medals for the United States between the two presented models. This may be a result of the host effect needing a more holistic view on a larger data set, as opposed to a linear relationship, which is why the ANN predicted the United States to receive more medals.

5.3 Relationship between Events and Medals

In Section 2.2, we found that the number of events a country participated in is statistically significant in predicting the total medals that country will win in a year. However, we consider this alone to be trivial, as a country will intuitively have a higher chance of obtaining medals if they participate in more events.

On the other hand, the "mean popularity" of the events each country participated in (`Mean.Event.Participation`) was relevant below a significance level of 0.05 for the total medal counts. The correlation was slightly negative—thus, greater event popularity leads to slightly smaller medal counts.

We did not observe a significant linear correlation between the latter metric and whether

the given country is the host (`Is.host`)—however, we did find a negative linear correlation in the interaction term between number of events and the `Is.host` flag and all medal counts. Curiously, this would mean that if a country is the host and participates in a large number of events, the model expects a decrease in the all number of medals obtained.

5.4 "Great Coach" Effect

We have found numerous examples of the *Great Coach Effect* as defined by organizer in the problem statement, and we noticed that this effect did bring more medals to the participating country. To find such an effect, we went through all historical data from the `summerOly_athletes.csv` file and find the countries which finished on the podium for each event for each year. After that, we compared the podiums in all events between each consecutive year, analyzing if the podium had changed. The program required a country not be in the podium at all in a given year, to then podium the following year (or within 3 Summer Olympics) in the same discipline. The below table shows a few examples of such. Full results may be found here.

Table 15: The existence of Great Coach effect among the participating countries

Name of the Country	Discipline	Great Coach Effect Interval
Cuba	Wrestling Men's Featherweight Freestyle	Between 1988 and 1992
Argentina	Tennis Men's Double	Between 1988 and 1992
Italy	Gymnastics Men's Team All-around	Between 1908 and 1912

6 Conclusions

In this model, we construct two models to predict the total score of each country. After weighing the results of these two models, we obtain an integrated result, as shown in the table.10, the United States will win the most medals, 131. then, the following countries, France, China, Italy, and Germany, will win 53, 40, 33, and 33 medals. If we count the number of gold medals gained by each country, then the result will change to the United States winning the most gold medals, 60, and then the following countries, France, China, Italy, and Germany, will win 15, 24, 19,14. We also found that many countries, such as France, Australia, and Japan, will perform worse in the 2028 Olympic games than they did in 2024. We found evidence of the effect of excellent coaches in multiple disciplines over the years.

7 Evaluation of the Model

Certain features of the models used in this paper work better for different questions one may have regarding the 2028 Olympics medal distribution. In truth, a combination of the ANN and MLR is useful for a more rounded overall model. While certain predictions are consistent across both models, there are crucial differences which make one better than the other in certain aspects. For example, the ANN model predicts that no

new countries will win a medal in the 2028 Olympics, and it is quite easy to understand why. The model has learned that over the span of 128 years and 32 Summer Olympic Events, a particular country has not won once. On the basis of that knowledge, there is no reason why it should suggest that suddenly a country may win a medal. In this case, the linear regression model reigns supreme, as it operates on an entirely different basis, as does not fall short to the situation described above. Conversely, finding relationships of specific variables, rather than entire datasets, the MLR model is able to more accurately predict whether a country will win its first medal based on its attributes. Another consequence of the ANN learning entire data sets is a possible erroneous prediction of certain team medals. For instance, China has taken part in merely 11 Olympics, as opposed to the United States which took part in almost 30. This results in the ANN assigning China 0 medals for all Olympics it had not participated in, reinforcing a notion that China's recent successes a merely a fluke, not replicable in future events. Again, in such situations the MLR has an advantage over the ANN. The ANN model works better for conclusions requiring a view on the whole set of data, such as the influence of *Host Advantage*. The MLR tries to account for this metric, but finds very little correlation, since it bases its predictions on only 1 Summer Olympic prior. This may explain the reason for a bigger discrepancy for the number of medals won by the United States than for any other country - perhaps the ANN found support for Host Advantage playing a role, and predicts the United States, as a host of the 2028 Los Angeles Olympics, will earn more medals. Many such differences exist between the two models, which is why we elected to construct both in harmony with each other. Their collective predictions should be a much more trust-worthy metric, as compared to both models in solitude.

8 Strengths and weaknesses

8.1 Strengths

- We weigh multiple approaches to get the final result about how many medals each country will gain. Thus, the results will be more accurate than those from a single method.

8.2 Weaknesses

- Gamma distribution might not be able to exactly describe the distribution of the medals earned of each country
- The ANN model takes into account all historical data, and sometimes predicts countries like the Soviet Union, Yugoslavia or Czechoslovakia as a possible winner. While the model ranks these countries very low, it ranks them nonetheless, meaning it may in theory predict that a non-existent country will win a medal, which is impossible.
- The MLR model suffers from non-constant error variance, also known as heteroscedasticity, which may lead to underestimated variance in the model.

References

Halsey, Lewis (Dec. 2009). "The true success of nations at recent Olympic Games: comparing actual versus expected medal success". In: *Sport in Society* 12, pp. 1353–1368. DOI: 10.1080/17430430903204892.

Hlavac, Marek (2022). *stargazer: Well-Formatted Regression and Summary Statistics Tables*. R package version 5.2.3. URL: https://CRAN.R-project.org/package=stargazer.

Moolchandani, Jhankar et al. (2024). "Predictive Analytics in Sports: Using Machine Learning to Forecast Outcomes and Medal Tally Trends at the 2024 Summer Olympics". In: *2024 4th International Conference on Technological Advancements in Computational Sciences (ICTACS)*, pp. 1987–1992. DOI: 10.1109/ICTACS62700.2024.10840553.

Appendices

Appendix A Modified Levene Test (Brown-Forsythe)

This test is performed to identify non-constant error variances. First, the linear regression model is fit, then residuals e_i are calculated. Then, we split sample into g groups, typically $g = 2$. Set group 1 to the residuals of the n_1 lowest values of the predictor X, set group 2 to the residuals of the n_2 highest values of the predictor X. We then set up the test as follows:

$$H_0 : \text{variance is constant}, \quad H_a : \text{variance is not constant}.$$

Calculate $d_{i1} = |e_{i1} - \tilde{e}_1|$ for group 1 and $d_{i2} = |e_{i2} - \tilde{e}_2|$ for group 2, etc., where e_{ij} is the ith residual of the jth group and $\tilde{e}_j$ is the median residual of the jth group. Finally, conduct a level-α two sample t-test with d_{ij} using the test statistic

$$t^* = \frac{d_{i1} - d_{i2}}{s\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}},$$

where

$$s^2 = \frac{\sum(d_{i1} - \bar{d}_1)^2 + \sum(d_{i2} - \bar{d}_2)^2}{n - 2}$$

is the pooled variance. We reject H_0 if $|t^*| > t\left(1 - \frac{\alpha}{2}; n - 2\right)$, where $t\left(1 - \frac{\alpha}{2}; n - 2\right)$ is the $1 - \frac{\alpha}{2}$ quantile of the student's t-distribution.

Appendix B GitHub Repository

All of the code behind our work, including the source code for this PDF, can be found at https://github.com/YanxiangShan/MCM-2524908.