FIFA´s player potential predictor with multiple linear regression

Abstract - This document provides documentation related to implementation, explanation, and comparison between a linear regression implementation by hand and sklearn own linear regression module to determine FIFA's player potential.

I. Introduction

With the recent inauguration of the Mexican eligaMX, which is a professional FIFA league that involves all the 18 teams that currently participate in the Mexican football league. The necessity of information and data analysis for the correct in-game player selection to develop in-game players and have a better team overall was born.

This project focuses on player potential, which tells us how much a player can improve with in-game training.

II. Dataset

A FIFA players attributes dataset was taken from Kaggle website contributed [1]by Sushova Patra an Indian data analyst.

This dataset has over 17981 player's attributes from all the national leagues around the world registered under FIFA license.

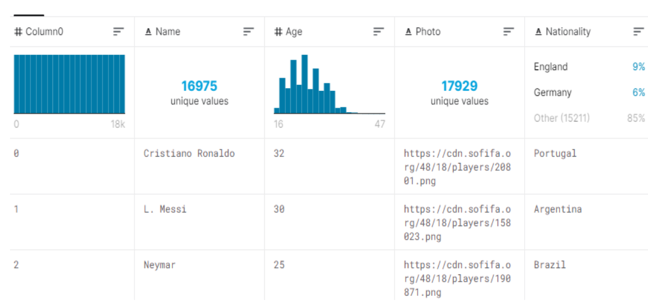

Figure 1. Kaggle dataset chunk

A preprocess of the dataset was needed in order to prepare it for a linear regression model. Data was filtered to keep only the relevant and essential features, for this a correlation table was used. In addition to the correlation, football and player

[1] https://www.kaggle.com/edith2021/fifa-18-player-prediction

development knowledge also discriminated features.

At the end, the features selected for the model were:

```
---  ------        --------------  -----
 0   Age           15952 non-null  int64
 1   Overall       15952 non-null  int64
 2   Potential     15952 non-null  int64
 3   Acceleration  15952 non-null  float6
 4   Agility       15952 non-null  float6
 5   Balance       15952 non-null  float6
 6   Ball control  15952 non-null  float6
 7   Reactions     15952 non-null  float6
 8   Stamina       15952 non-null  float6
 9   Strength      15952 non-null  float6
 10  Vision        15952 non-null  float6
```

Figure 2. Selected Features and Y

All related to mental and physical attributes to determine our Y, which is Potential.

	Age	Overall	Potential
Age	1.000000	0.462062	-0.228583
Overall	0.462062	1.000000	0.675489
Potential	-0.228583	0.675489	1.000000
Acceleration	-0.187989	0.171659	0.244556
Agility	-0.010481	0.251199	0.224871
Balance	-0.080641	0.061452	0.112374
Ball control	0.250885	0.704749	0.529906
Reactions	0.466492	0.837029	0.510583
Stamina	0.206724	0.455517	0.241944
Strength	0.340734	0.345676	0.089749
Vision	0.246319	0.508824	0.345994

Figure 3. Correlation table

III. Approach[2]

The objective is to predict the potential of the player according to attributes of all the players registered in the game, since we only have numerical values, this can be achieved by doing a linear regression.

In addition, the second objective is to compare results between a by hand implementation and the sklearn module.

The by hand implementation uses gradient descent algorithm, and sklearn module uses ordinary least squares.

Gradient Descent

$$\boxed{\Theta_j = \Theta_j - \underset{\text{Learning Rate}}{\alpha} \frac{\partial}{\partial \Theta_j} J(\Theta_0, \Theta_1)}$$

Figure 4. Gradient descent algorithm

$$y = \beta X + \epsilon$$

3 https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/

4 https://statisticsbyjim.com/glossary/ordinary-least-squares/

Figure 5. Ordinary least squares algorithm, where x represents the features and Beta the parameter to be estimated.

By hand implementation is based on the Linear regression from scratch from Levent Baş.

IV. Result

The training and testing accuracy (Variance) was used to determine how far our model matches our data. In this project variance values were:

- Training
    - By hand: `0.8381`
    - Sklearn: `0.8381`
- Testing
    - By hand: `0.8352`
    - Sklearn: `0.8360`

Which tells us that both implementations are almost the same.

To evaluate model skill to predict the potential of a player, this project used a cross-validation score, whose value was 84%.

In order to determine which prediction was better between the sklearn and the scratch model, mean squared error was calculated, whose value was 2.433.

V. Conclusion

As we can see, sklearn framework and by hand methods had a good performance overall, both had similar outcomes, even though they used diferente algorithms, performance was not affected, but we can say that the small amount of difference between both implementations is related to the extra tools that sklearn module has for these problems.

VI. References

[1] L. Bas, "Towards Data Science," [Online]. Available: https://towardsdatascience.com/linear-regression-from-scratch-with-numpy-implementation-finally-8e617d8e274c. [Accessed 31 05 2021].

[2] S. Patra, "Kaggle," [Online]. Available: https://www.kaggle.com/edith2021/fifa-18-player-prediction. [Accessed 31 05 2021].

[3] Wikipedia, "Wikipedia," [Online]. Available:

https://en.wikipedia.org/wiki/Variance. [Accessed 31 05 2021].

[4] S. Learn, "Scikit Learn," [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html. [Accessed 31 05 2021].