FIFA PLAYER RATINGS ESTIMATION

İsmail Onur Yılmaz
Arda Karabiber

Introduction:

FIFA has been one of the most popular sports games in the world for years. On the top of it, the game transformed itself into a e-sports game where there are lots of professional e-sports players compete in worldwide tournaments. The most important aspect of the game is the rating scores of players in teams. These overall scores are crucial to win the games, so it is fair to say that this is the most discussed feature of the game. Therefore, we decided to examine how overall ratings of players are calculated by using supervised machine learning models.

First, we acquired player database that includes a lot of information about the players including their overall rating. By using these features, we implemented several machine learning models that predict players' overall rating. Then, we compared the results with different score metrics.

For the second part, our aim was observing how are the player overall ratings parallel to their performance in real life. To achieve that, we acquired player statistics for the previous season from Transfermakt database which is one of the most credible public databases available. Later, we merged FIFA player dataset and this one to create a new model. This time, we used player statistics from Transfermarkt database to predict player overall ratings in FIFA. We implemented different models and analyzed their performances with different score metrics.

PART I

Data Processing:

The dataset is used for this project is FIFA players database that includes players' different features with their overall ratings. There are 19,239 players and 110 features in total in this dataset. Our target value "overall" feature has no missing values. Some features, like player name and jersey number, are dropped from the dataset since they have no significant effect on player overall rating. After this process, there are 54 features left.

Most of the players have more than one position under "player_positions" feature and that might cause unwanted problems while running the models. To prevent this issue, only their primary positions are acquired (the first position written in the related feature) while others are excluded. Still, there were too many unique categorical values and to reduce that, all positions gathered under 4 categories that are goalkeeper, defender, midfielder, and forward. Thanks to that categorization, identifying positions of players become simpler and more effective.

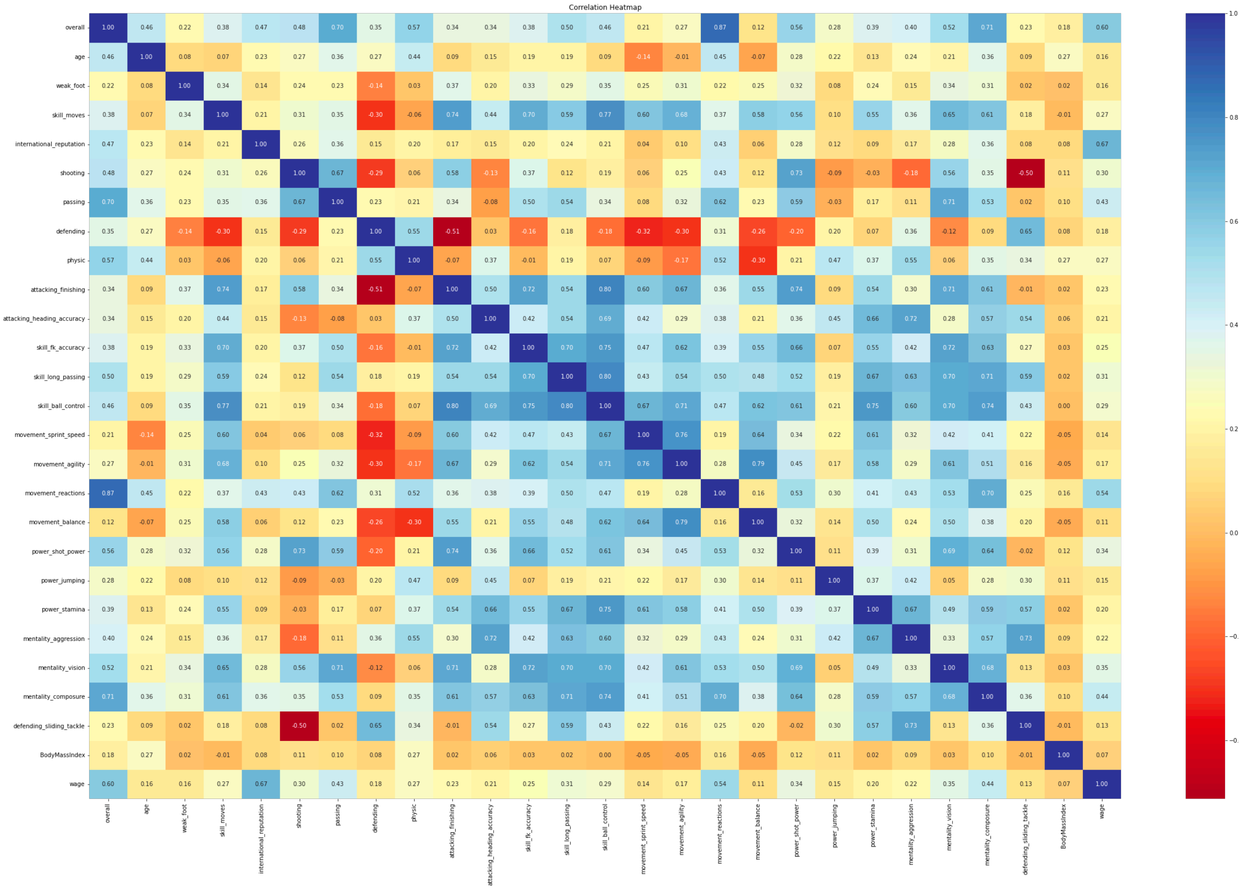
5 of the 54 features had 2132 missing values and these missing values were "shooting", "passing", "dribbling", "defending", and "physic". We suspected that can be a result of goalkeepers and when we checked, all of these missing values were from goalkeepers. To fix this issue, we matched 5 different goalkeeper attributes from the same dataset with these 5 features with missing values for the goalkeepers and dropped the goalkeeper features since they are not needed for non-goalkeeper players anymore. This will not cause a trouble since positions of players are already categorized, so goalkeepers and non-goalkeepers can be discriminated easily.

There were only 3 features with missing values left. These were "value_eur", "wage_eur", and "release_clause_eur". When the correlation between these features is observed, it is easy to notice that they are highly correlated. Thus, only one of them (wage_eur) is kept and the other two are excluded. Wage feature has only 60 missing values. For that problem, KNN imputation methos is used with k value of 5.

There were 3 physical indicator features that are weigh, height, and body type. A new column that calculates body mass index is added instead of these and these three columns are dropped. There was also another column that demonstrated work rate of players. However, it had too many unique categories, and since it had no visible impact on overall, it was dropped from the dataset.

Correlation between features is analyzed with correlation matrix and 14 of them are removed in order to prevent multicollinearity since they have high correlation. Final correlation matrix after removing highly correlated features is shown below:

Figure 1: Correlation matrix after removing highly correlated features



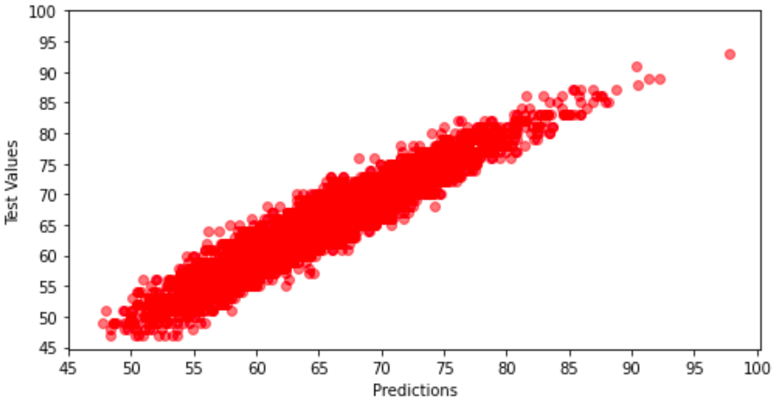
Dummy variables are used for “preferred_foot” and “Position_Group” features since they have categorical values. The final form of the dataset consists of 30 independent variables and 1 dependent variable with 19,239 datapoints. Dataset is split into test and train with test size of 0.25. Also, numerical features are scaled by using Standard Scaler. To prevent data leakage, train set is scaled with fit transform method, while test set is scaled with transform method. Lastly, scaled numerical features are added to categorical values and the dataset becomes ready for running the models.

Running Models:

For this phase, regression models are selected since the target variable is a numerical feature, not a categorical feature. Models used in this section are Linear Regression, Ridge and Lasso, Decision Tree Regressor, Random Forest Regressor, XGBoost Regressor, and LightGBM Regressor.

Linear regression model performed very well with r square of 0.9 and MSE of 4.5. Difference between test values and their predictions can be seen in a graph below:

Figure 2: Correlation of test values and predictions



Ridge and Lasso methods are implemented after linear regression. To observe the impact of different alpha values, grid search is used. Data is separated in 5 folds in this process. With that, we also applied a cross-validation process. Negative mean squared error is used for performance metric in this phase. Both Ridge and Lasso performed with almost identical MSE values of 4.6 which is also very close to MSE of linear regression model. That means linear regression model did not overfit the data.

After that, a decision tree regressor model is implemented. We did not use grid search but added some parameters manually to prevent the model to overfit with automatic parameters. Non-scaled data is used for this phase since scaling is not necessary for tree-based models. MSE score of this model is 6.1 which is significantly higher than previous models. It is fair to say that decision tree model underperformed in this setting.

Next model used is random forest regressor. Same process with decision tree is implemented for random forest. Parameters are written manually without a grid search. Model performed with 0.92 r square and 3.8 MSE. It is obvious that it is the best performed model until now.

Fifth model used on the dataset is XGBoost regressor. Since this model has too many parameters, we decided to use grid search to acquire better results. Parameters selected for grid search are "n_estimators", "max_depth", "learning_rate", and "reg_alpha". To prevent the model to overfit, "n_estimators" is kept lower and "learning_rate" is kept higher. "reg_alpha" is added for lasso penalization. Also, cross validation with cv=5 is used in this model. The model with best parameters performed best among all models with 1.12 MSE.

Last model used is LightGBM classifier. The same process with XGBoost is applied in this one. It performed well too, only slightly worse than XGBoost with 1.17 MSE.

Table 1: MSE scores of all models

Model	**MSE**
Linear Regression	4.5039
Ridge and Lasso	4.6361
Decision Tree Regressor	6.1673
Random Forest Regressor	3.8381
XGBoost Regressor	1.1247
LightGBM Regressor	1.1729

To sum up, all models performed high with low MSE values. Especially XGBoost and LightGBM regressors performed extraordinarily well. We can say that in-game player features play a considerable role while determining the player overall ratings.

PART II

Data Processing:

For the second part of the project, real life player statistics from previous season are acquired to predict their FIFA overall ratings. Our aim here is to see if FIFA ratings coincide with real life performances. The data is acquired from transfermarkt database. Dataset includes 2644 datapoints and 400 features in total. Among these 400 features, only most important and impactful ones are selected. These are the ones most likely to measure a player's performance in general. Also, there were several features that were very similar to each other like total goals and goals per 90 minutes. Thus, these features are eliminated, and total number of features decreased to 26.

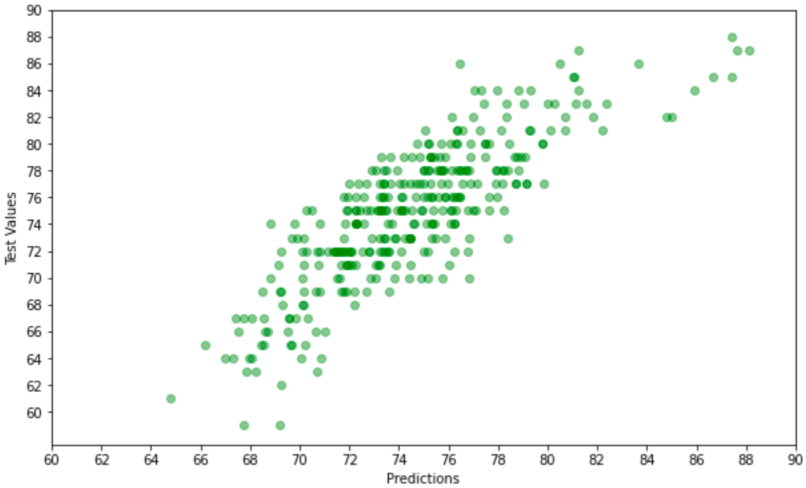
Next, two datasets (FIFA and transfermarkt) are combined into one dataset based on player names. To perform that, player names are transformed into lowercase and space between first name and last name are removed in both datasets to make matching the names easier. We could acquire 1661 datapoints which is equal to 63% of datapoints in transfermakt dataset with this procedure. It is not certain what happened to missing points, but it is possible that some of them are not included in both datasets. So, it Is expected to lose some of the datapoints in this phase. While combining the datasets, only "overall", "age", "BodyMassIndex", "preferred_foot" and "Position_Group" features are included from FIFA dataset since these are real features of players not fictional game features. At the end, the final dataset has 1661 datapoints with 31 independent variable and 1 dependent variable. Dataset is split into train and test with 0.2 test size.

Running Models:

Same models with first part are used in this phase since it still is a regression problem. For this part lower success rate is expected since ratings from a game might not fit with real life statistics perfectly.

First, linear regression model is implemented. This time, the dataset is not scaled since real life statistics that were chosen for the model are statistics per match and they are naturally scaled in that way. This model achieved a 9.2 MSE with 0.67 r square. Difference between test values and their predictions can be seen in a graph below:

Figure 3: Correlation of test values and predictions in second part



For Lasso and Ridge method, grid search is used for different alpha values with negative mean squared error as a performance metric. Again, cross validation with cv=5 is applied in this model. MSE of Lasso is 12.99, while MSE of Ridge is 13.59.

Next model implemented is decision tree regressor. Different from the one in first part, we used grid search this time in order to achieve a higher result. Selected features for grid search are “max_depth”, “max_features”, and “min_samples_split”. MSE acquired with best parameters is 13.7.

After that, a random forest regressor is implemented with a grid search. It performed better than previous models with MSE of 8.85

XGBoost with grid search is applied as the fifth model. It performed slightly better than random forest with a MSE of 8.81. The margin is almost insignificant in this case.

Last model implemented is LightGBM regressor. It achieved lowest MSE which is 8.39. In this case it is the best performed model among all 6 different models.

Conclusion:

We had only 1661 football players to use for the second part. Therefore, the performances of the models are limited with this amount of data. Also, more features could be acquired from transfermakt dataset might improve the results. Still, it is fair to say that FIFA overall ratings of football players are parallel to their real-life performances. We can say that even though FIFA is just a game, it is a realistic game based on player ratings.