Github URL

URL: https://github.com/PaulyCorcoran/UCDPA_PaulCorcoran

Abstract

The EA Sports FIFA football game series has been one of the most popular and cherished video games in the world for over twenty years. During the creation of this game, each year player statistics are intrinsically measured and updated in the game's database. Real world performances are reflected in the players individual and overall team rankings in the video game. The purpose of this project was to apply the learnings from the Specialist Certificate in data analytics to a real world dataset and aim to measure of the game's most important player variables.

Introduction

The first stages of the project involved the retrieval of the datasets for the past two iterations of the FIFA game series. The process of data cleaning and exploration is observed in steps 2-4. After which the modeling stages and hyperparameter tuning/gradient boosting takes place in steps 5-9. The players transfer value in euros was the first variable that the project aimed to predict. The project uses a linear regression to predict this variable based off the dataset provided. A secondary project goal of correctly classifying player positions (Defender, Midfielder or Forward) was established during the work of the project to provide an illustration of the method of classification modeling.

Dataset

The two dataset's were sourced from Kaggle from the following URL's using the Kaggle API.

FIFA 22 - https://www.kaggle.com/stefanoleone992/fifa-22-complete-player-dataset
FIFA 21 - https://www.kaggle.com/stefanoleone992/fifa-21-complete-player-dataset

Both datasets contained 107 variables ranging from player physical statistics should as height, weight, shooting and strength to mental statistics such as composure. Interestingly, the dataset contained continuous variables such as the players value in euro and the player's wage per week in euros. Originally, the variable "Overall" which is the players overall skill rating (Messi, who is the games best player is rated the highest at 93) was chosen to be modelled but due to the fact that this is a discrete variable (values range from 40-100), the variable was not suitable for a linear regression.

Implementation Process : Linear Regression Model

The dataset was merged into one complete data frame called 'fifa_complete_data' from the two sourced files. This initial merged file contained 38,188 rows and 107 columns. EDA was conducted on the dataset to discover that there was a high number of duplicate player rows, this was expected as the majority of player's would have been active in last year's game. Coupled with the addition of new players from the academy making their entry into the first teams and the addition of new teams our final dataset after the cull was expected to be large. A number of criterion was applied to the dataset to reduce the size and potential noise for later modelling. They are as follows:

1. **Drop all duplicate players but keep the latest game FIFA statistics for analysis –** this way we would have the most current player metrics for modelling.
2. **Remove all players with an skill level of less than 70 –** removing the lower skilled players would potentially reduce the variance observed. The lower skilled players transfer value was insignificant compared to the games stars.
3. **Recraft player positions and remove the rows with Goalkeeper as the position –** a dictionary was used to iterate over the player_positions column to categorise the players into "Defender", "Goalkeeper", "Midfield" or "Forward. The players who were categorised as a goalkeeper were missing a huge number of outfield statistics and would have damaged the modelling process and were dropped.

Before step 3 above the data frame consisted of the below player positions. Goalkeeper's consisted of just 10% of the player data so it was considered redundant and subjected to listwise deletion.

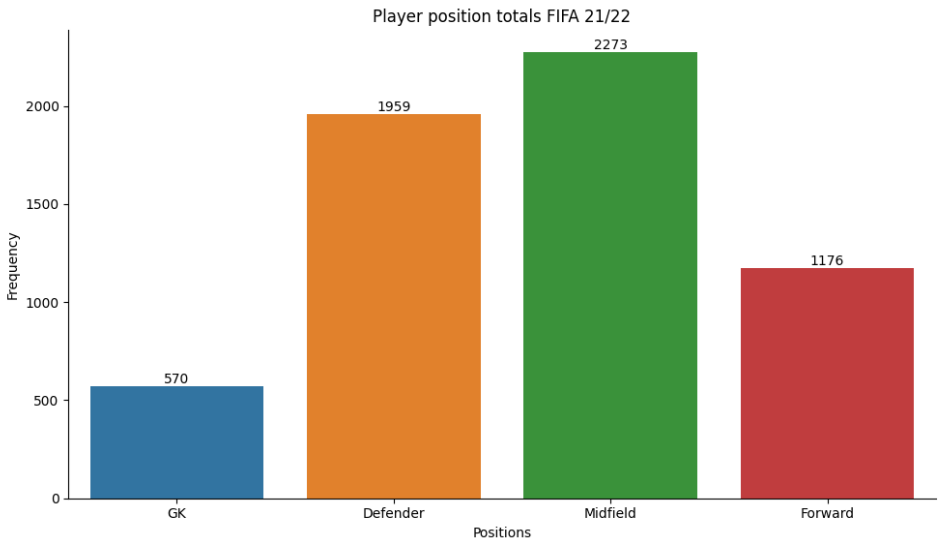

Figure 1: Player Positions in dataset

There was further dimension reduction in removing the vast amount of categorical and string variables. 50 variables were removed such as the player team name and image URL's concerning the players facial recognition for example.

Correlations

A Pearson correlation test was carried out on the target variable "Player_value_euros" to determine the variables in the dataset that correlated the most in the positive and negative direction. The results showed the following variables were the highest correlators.

1. Skill Rating – the players overall ability. (Discrete)
2. Release_clause_euro – the players release clause valuation. Correlated at 0.99. (Continuous)
3. Wage_euro – the players weekly wages. (Continuous)
4. Movement_reactions – the players physical stats for reactive ability (Discrete)
5. International_reputation – coded as 1 being the largest and 5 the lowest. (Discrete)

At this point in the project some alterations were made to the plan. The model R2 score with the continuous variables were extremely high meaning they more or less explained the target variable on their own. It was decided that a number a key indicators such as the continuous variables above and the "potential" variable which was highly linked to the players overall ability be dropped and the project be taken forward as if those variables were not available. Histograms were produced to evaluate the distribution of the variables which were chosen by evaluating the pearson plot which showed the correlation between the data set and the target variable. After this, scatterplots were produced to illustrate the linear relationship between predictor and dependent variables.

The variables chosen for the linear model were:

- Skill_rating
- Age
- Dribbling
- Shooting
- Movement_reactions
- Skill_ball_control
- Mentality_composure
- Attacking_short_passing

Scaling

Prior to the feature choice an attempt was made to scale the high variance variables such as the high value continuous variables. The effects of this scaling were evaluated through histograms and modelling which was later removed from the project as the effect of the scaling had no effect on the model performance.

Linear Regression model Results

Compared to a perfect regression line (figure 3) we can see the model performed reasonably well at predicting the transfer value of the majority of the dataset. The R2 score which accounts for how much variance is explained by the model clocked in at 70%. The root mean squared error was 6877752. The model achieved the same training and test score of 0.7% which means it could generalise fairly well to unseen data. The output of the function linear_reg() provides a full summary breakdown as well as the Feature importance breakdown.

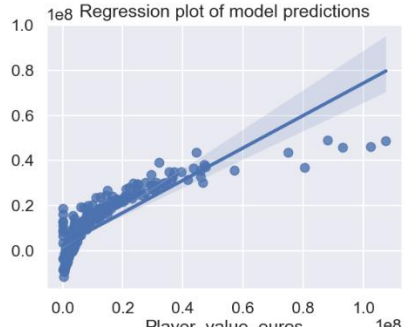

Figure 2: Model Linear Regression

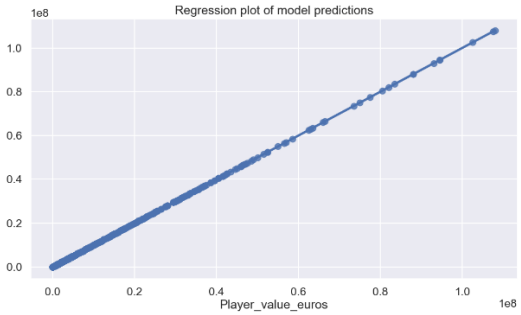

Figure 3: A perfect Linear Regression

Ridge optimisation and GridSearchCV were performed in order to improve model score, the gridsearch found that the optimal parameters for the ridge model to be # {'alpha': 0.0001, 'fit_intercept': True, 'solver': 'saga'}. Unfortunately, this yield no improvement to training/test scores and the RSME was clocked in at a highly similar 6877927.

Implementation Process : Player Position Classification Model

During the cleaning and EDA phase of the project I described earlier how the player positions was recrafted to 3 main categories. This provided the perfect platform to create a classification model to evaluate how well we could predict a players position given the vast physical and mental statistics at our disposal. The dt_dataframe was created by removing all string/categorical variables like before and by reclassifying the player_positions variable to be {Forward : 0, Midfield : 1 and Defender : 2}. This allowed for a classification model such as a decision tree to be modelled on the dataset. Again, a pearson correlation plot was utilised to great effect, the strongest correlating variables were discovered. It was decided however, that after testing of this condensed modelling dataset (6

variables) that the approach here was to run the whole dataset through the model to achieve the best result.

Classification Model Results

The initial result of the DecisionTreeClassifier is as follows. The Accuracy for this model was 83% which was a good starting point for further optimisation.

	precision	recall	f1-score	support
0	0.72	0.78	0.75	324
1	0.82	0.78	0.80	696
2	0.90	0.92	0.91	603
accuracy			0.83	1623
macro avg	0.81	0.82	0.82	1623
weighted avg	0.83	0.83	0.83	1623

Accuracy: 0.83

Figure 4: Decision Tree Classification report.

Ensemble methods

Ensemble methods work by combining models that perform the best, The group (or ensemble) will regularly perform better than the best individual model, especially if the individual models make different types of errors. (Geron, 2019)

The RandomForestClassifier was initialised and through the function created model_tuning_GS the optimal parameters were found for the ensemble method in order to improve accuracy. The parameter_dict was a dictionary of all the possible parameters that I wished to test on the model. The GridSearch function applied each of the parameters in the testing of the RF model and through the results of this the best parameters were found to be 100 estimators and a max depth of 5. By optimising the ensemble model Accuracy was found to have increased 3% to 86%.

Gradient Boosting

"Boosting (originally called hypothesis boosting) refers to any Ensemble method that can combine several weak learners into a strong learner. The general idea of most boosting methods is to train predictors sequentially, each trying to correct its predecessor." (Geron, 2019)

Gradient Boosting, one of the more popular techniques for squeezing extra accuracy from models works by sequentially adding predictors to the ensemble, each new predictor seeks to correct its

predecessor. Just like the RF optimisation of parameters the function model_tuning_GS was applied to the GradientBoostingClassifier with a dictionary of parameters to optimise. The results of that optimisation showed the best model parameters to be: (learning_rate=0.01, max_depth=5, n_estimators=200, random_state=42) The Gradient boosting of the classification model allowed for a further increase of 3% to 89% accuracy.

Insights

- The majority of Physical and mental statistics such as attacking_short_passing, mentality_composure & movement_reactions were found to be normally distributed, that is to say that the players analysed were closely gathered around the mean in each statistic, this was not the same in player_value as some players worth heavily skewed the data.

- The release_clause_euros of a player correlated so heavily with the players worth that a simple one variable regression would have been the best predictor, accounting for 96% of the variance. It generated scores of 100% on the training set and 90% on the test set.

Figure 5: one variable model

- Preferred_Foot, that is choice of either a player is left footed or right footed actually had a negative effect on transfer value. The variable was recrafted to {Left: 0, Right:1} so this shows as the variable increased the value went down. IE left footed players were on average more valuable.

- During the classification of player positions the variables found to be the biggest predictors were the defensive statistics. In other words it was easier to classify forwards as having deficiencies in these metrics and defenders having strength in these areas. Midfielders were found to have a mix of all around ability which would tally into natural thought about the game.

```
The variables that correlate with the target variable Player postions are
 player_positions              1.00
defending                      0.76
mentality_interceptions        0.73
defending_marking_awareness    0.74
defending_standing_tackle      0.75
defending_sliding_tackle       0.78
Name: player_positions, dtype: float64
```

Figure 6: Highest correlating factors in player positions.

- Player Positions was found to negatively correlate with a players transfer value, as the values went up {Forward : 0, Midfield: 1, Defender:2} this was found to reduce transfer value. This ties in with contemporary thinking as Forward players are worth more than defenders etc.

- Parameter tuning and boosting of the Classification model increased its accuracy by 6%.

References

Geron, A. (2019). *Hands on machine learning with Scikit-Learn, keras and tensorflow.* Beijing: O'Reilly. 2019