CS230

FIFANet: Deep Learning to Predict Player Value

Zach Taylor
Department of Computer Science
Stanford University
ztaylor@stanford.edu

Conner Smith
Department of Computer Science
Stanford University
csmith95@stanford.edu

Jonathan Tynan
Department of Computer Science
Stanford University
jetynan@stanford.edu

Abstract

In soccer, there is value in being able to predict how much a player will improve or decline from year to year, and how much a player should be valued on the transfer market. FIFA, a soccer video game, provides proxies for actual player skill through their detailed rating system. The motivation for our project is to be able to make inferences about the value and future performance of professional soccer players. We train and optimize deep learning models and standard machine learning models for two different tasks. First, we predict a player's monetary value in a given year using their attributes from that year. Second, we classify players based on whether they will improve or decline in rating in the next year. We report on the architecture, optimization, and performance of our models. Such value and performance predictions could potentially be used by soccer clubs to make informed transfers, by fantasy soccer fans to draft the right players, and by scouts and coaches to learn more about how players develop.

1 Introduction and Motivation

The motivation for our project is to utilize deep learning to make inferences about professional soccer players. There are two main patterns we investigate. First, we want to examine correlations between player attributes and the player's monetary value. The insights from this analysis can be used to identify the most impactful features that vary across positions. Additionally, it could be used to help gauge a player's marginal value to a team. Second, we leverage datasets spanning from 2017 to 2019 to explore a player's performance over time. We hope to uncover underlying patterns that influence a player's improvement or decline. Accurately determining player value is what clubs do to ensure they get the best deals possible on the transfer market. Additionally, forecasting player improvement allows for more informed decision making that takes into account the performance of the team in future years in addition to the current one. As a proxy for actual player skill, we utilize FIFA player catalogues from the past three years to build models capable of predicting player trajectories.

2 Related work

Soccer analytics research focuses on real-player data; consequently, there has not been much work conducted on the FIFA datasets. Nevertheless, one team created a multilayer-perceptron neural network to predict the price bracket of players from the FIFA 2017 dataset [1]. They classified each player into 1 of 119 price brackets. Their overall accuracy was 40.32%, but 87.2% of predictions were 2 price brackets or fewer away from the true

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

bracket. Additionally, they place any player within 6.32% of his actual price on average. We thought this was a clever way to transform a regression task into a classification task, though in our opinion, a regression task is the more natural expression of the problem.

Another team created a support vector regression model to predict the overall rating of players in the FIFA 2018 dataset [2]. They achieved a mean squared error of 1.23 and a variance score of 0.97, which seems to be state of the art in this particular task. Our task differs in that we have different outputs, but many of the features and methods are the same.

One published paper we found uses proprietary data from a sports analytics company to predict soccer player market value [3]. The median relative difference between their predictions of market value and actual market value was 34%, and the mean difference was 60% (showing the influence of outliers on the mean error measure. This paper, however, does not consider neural network models.

Existing work also exists using very different sources of data to analyze player performance. Computer vision techniques can be used to caputre player performance, segment players, quantify distances covered, and project player trajectories, among other applications [4]. This paper offered a good overview of the many different applications of vision to player performance, many of which seemed to provide good accuracy. We found less work, however, translating these vision approaches to season-to-season analyses.

Finally, we looked through a review paper on big data and tactical analysis in elite soccer [5]. This discussed the current work as well as the current limitations of using large datasets to capture team tactics. For example, distributed systems and modern big data techniques may be used to more accurately capture performance. This is different from the dataset that we have, however, which is on the order of tens of thousands of examples rather than hundreds of millions or larger.

In addition to using them as comparisons against which to evaluate our own model's performance, we hope to understand the research that went into these models to avoid potential pitfalls and gain insight into choosing the best structure and hyperparameters for our models.

3 Dataset and Features

3.1 Overview

Our dataset consists of 2017, 2018, and 2019 player catalogues from Kaggle. In each catalogue, every FIFA player is associated with a set of features. We labeled each player in 2019 as improved if their overall rating increased from 2018 to 2019 and not improved otherwise. Our dataset contains 36 features total, ranging from skill-based features such as overall rating, speed, stamina, passing accuracy, etc. to metadata such as position and age. All of the features are raw inputs to predict the log of market value, and all except overall rating are used to predict player improvement.

According to EA Sports, "a network of over 9000 members review the player's abilities, watch him play, and help assign him various ratings" [4]. Among the reviewers are coaches and professional scouts. Thus, we believe the quality of the assessment are reasonably reliable and can be used to extract meaningful information.

We preprocess by creating a naming fix scheme to address changes in column names accross years, narrowing down to the features that are common across years, encoding position, removing duplicates, and otherwise cleaning small issues in the ata. We normalize by fitting an sklearn StandardScaler on only the training data (to avoid leakage), and apply that to the training, dev, and test data. Due to space and the size of the labels of the vectors, we will leave the reader to visit the dataset pages [6, 7, 8] to visualize a training example.

3.2 Player Improvement Classification Task

Our definition of improvement is that a player improved iff his "Overall" attribute improved between the two years. For this classiication task, the train set consisted of 8898 players (all from 2017-2018 trajectory) while the dev/test sets consisted of 4449 players, drawn from 2018-2019 trajectory. 3194 players improved in the train set while 1911 and 1929 improved in the dev/test sets, respectively. Train set mean was 0.35 with a variance of 0.23 while the test set mean was 0.43 with a variance of 0.25. This means more players improved between 2018-2019 than 2017-2018.

3.3 Player Value Regression Task

The value regression task is defined as predicting a floating point value representing the log of the player value. This is to reduce the effect of large outliers in cost, as some players are worth over 100 million while many others are worth only thousands. For this task, the train set consists of 80% of the 8898 players from the 2019 dataset, and the dev and test sets are the even split of the remaining 20% (10% each).

3.4 Limitations

One drawback to using these three datasets is that the features, as well as the players, present vary from year to year. The most dire example is that the 2017 set doesn't include unique player IDs or many of the other skill-based features present in the 2018 and 2019 sets. As a result, in the process of cleaning and synchronizing the datasets, we pruned from roughly 16k to 9k players.

4 Methods

Here, we detail our methods. We will be restricting our coverage of the underlying theory behind the baseline models to short one line descriptions, since CS 230: Deep Learning, for which this paper is written, is primarily focused on the theory of deep learning specifically. We'll go slightly more in depth into the deep-learning-related algorithms.

We used the cross-entropy loss function as our loss function for classifying players as improved or non-improved:

$$L = -\sum_{i=1}^{m} y^{(i)} \log(\hat{y}^{i}) + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})$$

For our regression task, we used the mean squared error of our predictions of the log(player value) as our loss function.

4.1 Player Improvement Task

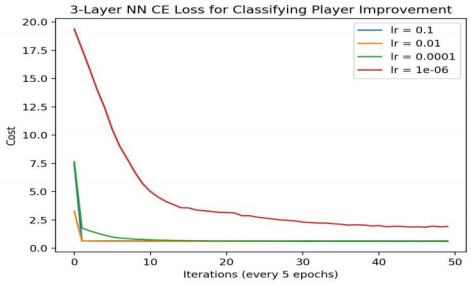
For our player improvement task, we used a variety of learning algorithms. As a baseline, we used a naive prediction of the mode of the training dataset- that is, for every player, we predict that they did not improve. Additionally, we trained logistic regression, which linearly combines input variables and applies a sigmoid output function to arrive at predictions [9]. We also trained a support vector classification model, which fits a hyperplane decision boundary based on the support vectors in the training set [10]. Additionally, we trained a random forest classifier, which uses an ensemble of decision trees that collectively contribute to a final classification decision [11]. Finally, we trained a variety of deep learning architectures, all in the form of standard feed-forward networks with 3 or more layers (which combine linear transformations with non-linearities to be able to fit complex functions), Xavier initialization to avoid equal update problems and exploding gradient problems, Relu activation functions, and sigmoid cross entropy loss or squared error depending on the task. The loss and accuracy graphs in the next section reflect the architectures and hyperparameters we explored, including BatchNorm, Dropout, Regularization, and learning rate adjustment. All NN architectures used Adam optimizer for stochastic gradient descent.

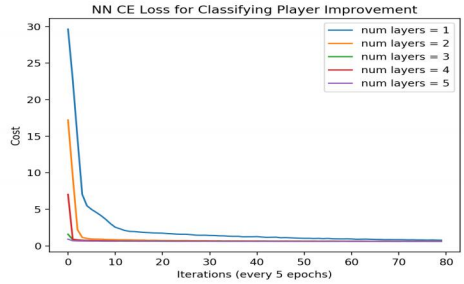
4.2 Player Value Prediction Task

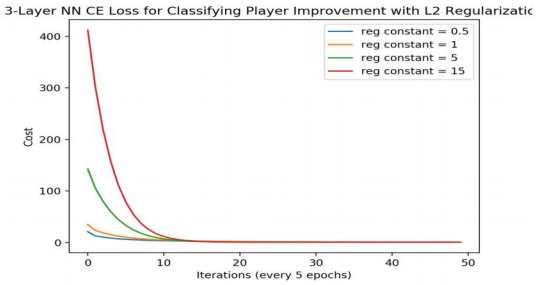
For our value prediction task, we used the following algorithms. First, our naive baseline consisted of always predicting the mean player value from the training set. Next, we used a linear regression model, which calculates the weights of a linear combination of the input variables and outputs the corresponding prediction on unseen examples [9]. Next, we trained a ridge regression model, which incorporates regularization into the cost equation [12]. Next, we trained regression versions of the support vector machine and random forest models mentioned previously. Lastly, we trained a variety of neural network architectures for the regression task.

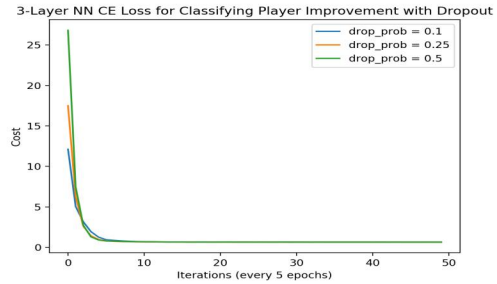
5 Experiments/Results/Discussion

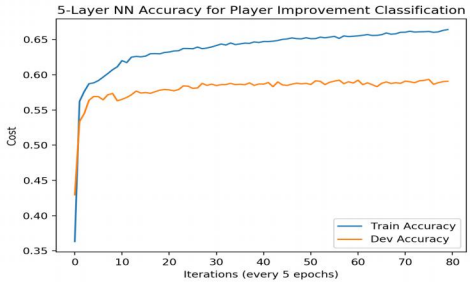
5.1 Player Improvement Task

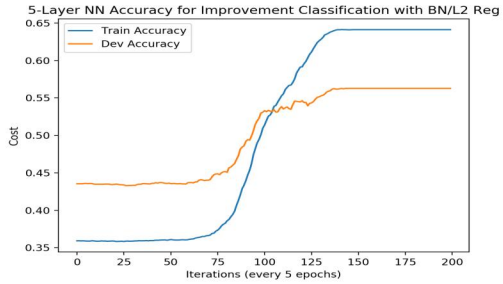


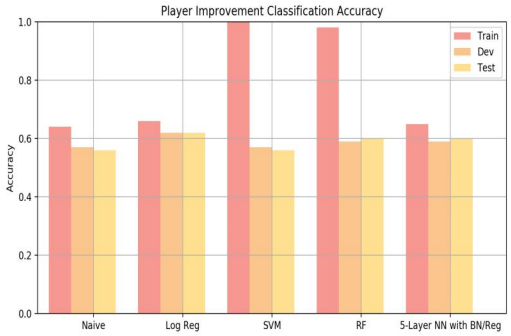


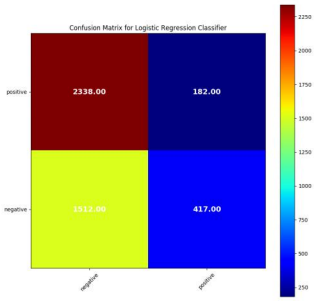












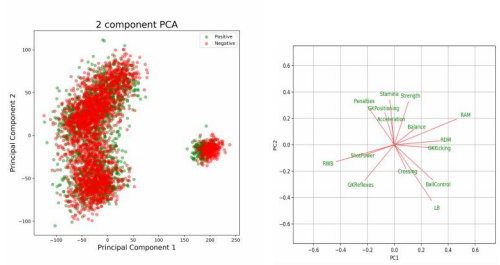
The best-performing NN was a a 5-layer NN (learning rate 0.0001 using Adam Optimizer for SGD) with BatchNorm and L2 regularization. BatchNorm helped by transforming the input signals to have particular mean and variance, which is helpful since some features, such as skill, have different scales than features like age. The marginal impact of 5 years is much greater than the marginal impact of 5 skill points, and BatchNorm helps the model adapt to these signals. L2 Regularization was obviously meant to combat overfitting on the train set. We chose a batch size of 128 since it gave good speed performance, and found that the best learning rate was 0.0001, since as seen in the plots, it optimizes the loss quickly, and it also achieved good test performance (though about the same as the larger learning rates).

The graphs above show that we experimented with different NN architectures and added optimizations like BatchNorm, Regularization, and Dropout. We experimented with deeper models (7, 9, and more layers) and larger and varied combinations of hidden unit sizes, and the best performing had 5 layers with hidden units sizes of 25, 25, 12, 12, 1 (plus biases). Our deep models didn't suffer from drastic overfitting, and regularization/dropout did a bit to combat the slight overfitting we observed. Some of our baseline models like SVM and random forest display significant overfitting, but their performance on the test set was on par with the other models. In the end, all of the models did at least as well as the naive baseline, but the "complex" models yielded no improvement over logistic regression. Logistic regression performed best on the test set, with an accuracy of 62%. The best NN architecture achieved 57% accuracy.

Below we list our classifier performance metrics. Here and in the other task, we list the best performing model, or the best performing model of each layer number for space (we tried many different hidden layer sizes and algorithm modifications, and found that eventually, adding layers resulted in diminishing returns):

Model	**Precision**	**Recall**	**Accuracy**
Naive	0.00	0.57	0.57
Logistic Regression	0.62	0.63	0.63
SVM	0.33	0.57	0.57
Random Forest	0.59	0.60	0.60
5-layer Neural Network	0.32	0.57	0.57

To better understand why this classification problem was so difficult, we created a PCA plot (below to the left) and its corresponding loading plot (below to the right) from the test set data:



In a qualitative analysis, we can see that there are two distinct groups in the PCA plot. The group on the left has a large variance in the second principal component, while the group on the right has very little variance in either component. However, neither group seems to correspond to the positive or negative labels, which may contribute to the difficulty of this classification problem. To better understand these groups, we created a loading plot. The positions of the players seem to influence the first principal component more and skill-based features seem to influence the second component more. However, this trend does not seem to correspond to any general classifications in soccer that would explain the PCA plot.

5.2 Player Value Prediction Task

Due to the 5 page limit, we are not able to include the same graphs here as in the previous section, but the basic trends and decision making processes are largely the same. Again, we experimented with different learning rates, batchNorm, regularization, and dropout, and found that our best performing neural network model was a 5 layer neural network with hidden layer sizes of 50 each, no dropout (which didn't affect performance much), batchNorm, and a learning rate of 0.0001. :

Model	**Mean Squared Error (after log transformation)**
Naive	3.81
Linear Regression	1.95
Ridge Regression	1.95
SVM	1.95
Random Forest	1.69
3-layer Neural Network	1.74
5-layer Neural Network	1.68
>= 7-layer Neural Networks	1.68

We found qualitatively that for the regression task, the model performed most poorly in accurately predicting the most expensive players. For some it gave good predictions, but when it didn't, it missed badly. All of the most expensive misses were on high values players, and particularly young ones, which makes sense because it is rare for a young player to be incredibly valuable, and much of their value is often based on their potential.

For example, Neymar (26) and Mbappe (19), valued at 41 million and 58 million respectively, were both in the top 10 worst misses. The mean squared error was heavily influenced by outliers, but in general we saw that the predictions were relatively good qualitatively.

6 Conclusion/Future Work

For the player improvement task, it was disheartening to see that the top performer was vanilla logistic regression. For the player market value regression task, we had better results, with a 5-layer NN outperforming all other models.

We believe there's great potential for applying deep learning to sports analytics. One idea is to investigate the PCA plot (section 5.1) further. The clustering suggests some interesting patterns hidden in the data, but our specific project goals failed to uncover/capitalize on these trends. The PCA clustering clearly doesn't correspond to player improvement, but perhaps this dataset can perform very well on classification tasks that capitalize on the clustering. Future work should prioritize gathering data that spans more years with the intention of building an RNN that can capture trends over time. Other projects could analyze data drawn from player statistics (versus FIFA virtual rating system).

7 Contributions

Conner performed data cleaning/preprocessing and developed much of the player improvement classification models then summarized the results in the poster and report.

Zach developed and tuned much of the regression classification models, researched related work, and worked on the motivation, related research, methods, and results/discussion sections of the report.

Jonathan created the PCA and loading plots and analyzed them in the report. He also contributed to the baseline models, researched related work, and worked on the poster.

References

[1] Dey, Sourya. "Pricing Football Players Using Neural Networks." ArXiv.org, 18 Nov. 2017, www.arxiv.org/abs/1711.05865.

[2] Lopez, Waldemar. "Applying Machine Learning to FIFA 2018." UruIT, 5 Feb. 2019, www.uruit.com/blog/soccer-and-machine-learning-tutorial/.

[3] R. Stanojevic and L. Gyarmati, "Towards Data-Driven Football Player Assessment," 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW), Barcelona, 2016, pp. 167-172.

[4] D'Orazio, T., and M. Leo. "A Review of Vision-Based Systems for Soccer Video Analysis." Pattern Recognition, vol. 43, no. 8, 2010, pp. 2911–2926., doi:10.1016/j.patcog.2010.03.009.

[5] Rein, Robert, and Daniel Memmert. "Big Data and Tactical Analysis in Elite Soccer: Future Challenges and Opportunities for Sports Science." SpringerPlus, vol. 5, no. 1, 2016, doi:10.1186/s40064-016-3108-2.

[6] Gadiya, Karan. "FIFA 19 complete player dataset." Kaggle. 21 Dec. 2018. 17 May 2019 <https://www.kaggle.com/karangadiya/fifa19>.

[7] Shrivastava, Aman. "FIFA 18 Complete Player Dataset." Kaggle, 30 Oct. 2017, www.kaggle.com/thec03u5/fifa-18-demo-player-dataset.

[8] Agarwal, Soumitra. "Complete FIFA 2017 Player Dataset (Global)." Kaggle, 13 Apr. 2017, www.kaggle.com/artimous/complete-fifa-2017-player-dataset-global.

[9] http://cs229.stanford.edu/notes/cs229-notes1.pdf

[10] http://cs229.stanford.edu/notes/cs229-notes3.pdf

[11] http://cs229.stanford.edu/notes/cs229-notes-dt.pdf

[12] http://statweb.stanford.edu/ tibs/sta305files/Rudyregularization.pdf