

Master of Technology (IS)

FIFA 19 Classification Project

Project Report

Team Member:

CAO LIANG A0012884E

GENG LIANGYU A0195278M

Contents

[Executive Summary 3](#_Toc19355003)

[Business Problem Background 3](#_Toc19355004)

[Project Objectives & Success Measurement 3](#_Toc19355005)

[Project Solution Design 4](#_Toc19355006)

[Project Files Structures 5](#_Toc19355007)

[Project Implementation 6](#_Toc19355008)

[1. Row Data Collection 6](#_Toc19355009)

[2. Pre-Processing 6](#_Toc19355010)

[3. Sampling 8](#_Toc19355011)

[4. Training Learning Algorithms 11](#_Toc19355012)

[Project Performance & Validation 20](#_Toc19355013)

[Project Conclusions: Findings & Recommendation 21](#_Toc19355014)

# Executive Summary

FIFA, also known as FIFA Football or FIFA Soccer, is a series of association football video games or football simulator, released annually by Electronic Arts under the EA Sports label. FIFA 19 data set comes from “<https://www.kaggle.com/karangadiya/fifa19>” which gets player information from “<https://sofifa.com/>”.

This project utilizes the several sklearn classifiers algorithms to predict the FIFA 2019 player international reputation scores.

# Business Problem Background

EA Sports’ expensive and luxurious FIFA 19 is undoubtedly football games’ equivalent of the Premier League. The international reputation values the football game player, and it is important factor to consider when choosing game player. Therefore, the project provides ways to predict and realise the importance of international reputation score.

# Project Objectives & Success Measurement

We will utilize existing FIFA 19 dataset and predict international reputation scores with selected relevant feature set.

The measurement for the project will depends on the prediction accuracy on the test data, and it should achieve 80% average accuracy or more to be success.

# Project Solution Design

The project should prepare the dataset with sample size greater than 3000, then select the sklearn classifiers and conduct the training, finally perform the verification of the training result according to average accuracy.

The project follows the supervised learning process as below.

# Project Files Structures

Current project contains following files:

1. conda-install.txt, packages which are needed by this project.
2. classification-data.ipynb, jupyter file which contains all the algorithms being used.
3. prepare-data.ipynb, the code we used to clean up and prepare the ready-to-use data.
4. /py\_src, contains all the python source files, which can be run from PyCharm or Spyder, common\_functions.py contains the functions which are shared by all the other files.

The other files are named by according algorithms used inside.

1. fifa19\_ready\_data.csv, processed data which is being used by all the algorithms.
2. fifa19\_data.csv, raw dataset gets from Kaggle.

# Project Implementation

The project performs the steps below.

## Row Data Collection

The Row Data “data.csv” contains 18208 records, and we select the records randomly to get 4000 records, so that there are extra records which can be filtered during pre-processing stage, and we still can meet minimum 3000 records requirement for the project.

## Pre-Processing

There are 4000 records with 89 features in the beginning of this stage, i.e. shape (4000, 89). We have spent lots of efforts to perform the following to process the records, so they are ready for training.

* 1. Removing Invalid Data

We perform this step so the overall data accuracy can be higher.

* + 1. Since we have set the target on “International Reputation”, so the records with missing “International Reputation” value will be removed.
    2. The scores for the root player positions (i.e. 'RAM', 'RB', 'RCB', 'RCM', 'RDM', 'RF', 'RM', 'RS', 'RW', 'RWB', 'ST') must exist. So if these score values are missing, the records also are removed.

The root positions are chosen with the following reason.

Since the player position left, centre and right are all considered the same score, so we pick only right player position to represent all positions (left, centre and right), and the right player position is called root position.

After 2-1 step, the records shape is reduced to (3563, 89).

* 1. Removing Features

We perform the steps below to increase the accuracy and also reduce the processing since the data shape will be reduced a lot.

* + 1. ‘Row\_number’ is removed since it only reflects the original data row number, and will not have any meaning for the project.
    2. The following features are removed since they are not used in the project.
* 'Name', 'Photo', 'Flag', 'Club Logo', 'Real Face', 'Jersey Number', 'Joined', 'Loaned From', 'Contract Valid Until', 'Release Clause'
  + 1. The following features are removed since we consider to choose the features without any bias. E.g. the international reputation score should not be related to Nationality, so people from different countries will be treated fairly regarding to international reputation.
* 'Nationality', 'Club', 'Preferred Foot', 'Body Type', 'Position', 'Weak Foot'
  + 1. The following player positions are removed since they have been represented by the root player positions.
* 'CAM', 'CB', 'CDM', 'CF', 'CM', 'LAM', 'LB', 'LCB', 'LCM', 'LDM', 'LF', 'LM', 'LS', 'LW', 'LWB'

After 2-2 step, the records shape is reduced to (3563, 57).

* 1. Converting Values

The values inside the data set contain many invalid data, such as empty value and string value with different format. We perform the steps below to convert the string value to numeric values, and also fill in the adequate numeric value for missing value.

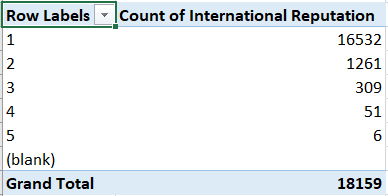
* + 1. Convert market and wage to the numeric value, and fill in mean of market and wage values individually in case of missing value.
    2. Convert height value from mixed inch and foot string value to the numeric value in foot only. Also fill in missing height value with mean height value.
    3. Convert weight value from string to numerical value, and fill in missing value with mean weight value.
    4. Convert position string value to numerical value.
    5. Convert work rate value to separate attack value and defence value in number. Work rate value is string value which contains attack rate and defence rate category, e.g. Medium/ Medium, and it is converted to numerical value, i.e. High is 3, Medium 2, Low 1.
    6. Convert the following string values to numeric values, and fill in mean value on the missing value.
* 'Age', 'Overall', 'Potential', 'Special'

After pre-processing, the data set shape is (3563, 58), and is reduced a lot compared to original shape (4000, 89). Also it meets the project requirement for at least 3000 records with 30 features.

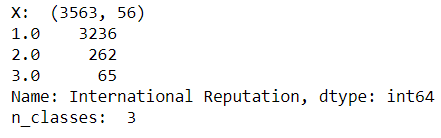
## Sampling

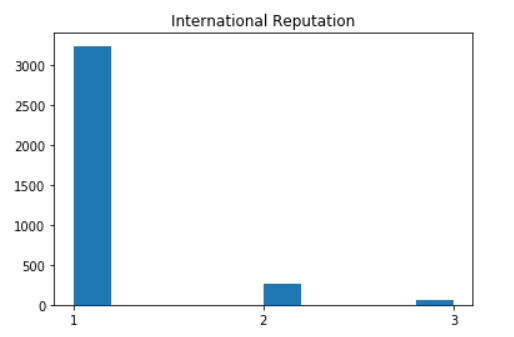
* 1. Adjusting Rating Score

Since rating score 5 has only 6 records within all 18159 records, and rating score 4 has only 51 records, we choose the re-assign the records with rating score 5 or 4 to be rating score 3, so the accuracy will be reasonable.



After that, the international reputation scores distribution is shown below.

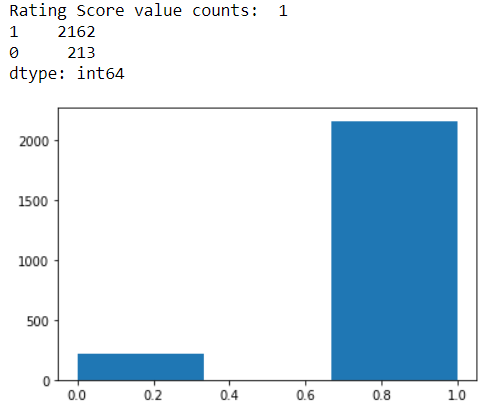


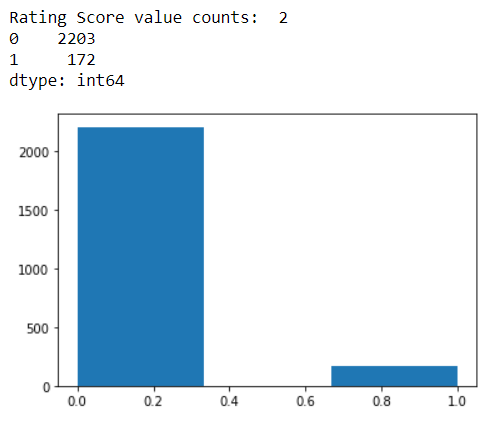


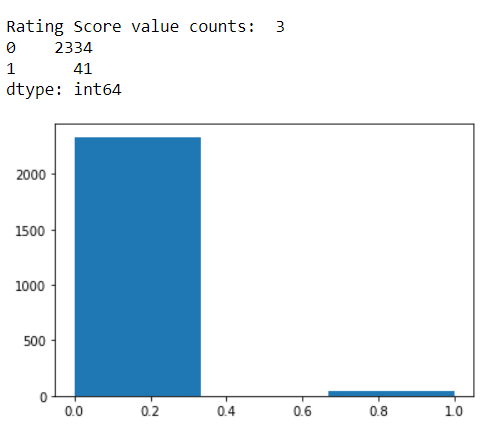
* 1. Converting Rating Score Label to Binary Categories Values

We utilize ‘label\_binarize’ function to convert rating score values to 3 binary categories values. The data distribution for different rating scores are shown below.

This step will enable the one vs rest classifiers at training stage can be utilized. And from our experiment, this improves the accuracy result significantly, when comparing the way if we create 3 different classifiers in same type and then train them.







* 1. Splitting Training Data Set and Testing Data Set

We split data set as 2/3 for training, and the rest 1/3 for testing.

* 1. Scaling Data

We scale data with ‘StandardScaler’ before training.

## Training Learning Algorithms

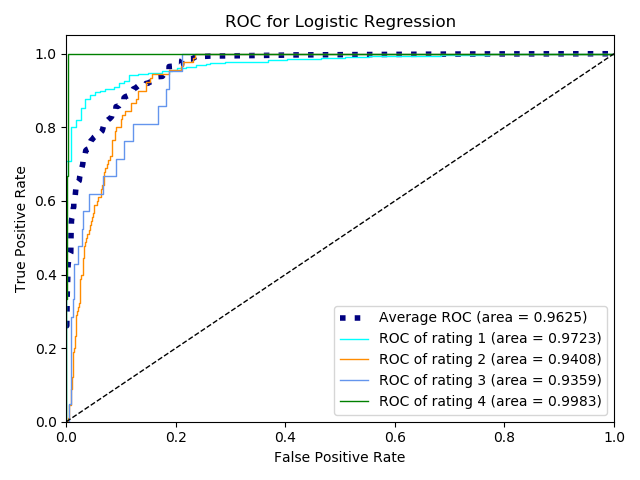
We have implemented 6 classifiers to deal with the classification problem.

* 1. Logistic Regression

We choose the option multi\_class='ovr', and also use OneVsRestClassifier to improve the overall accuracy.

After optimization with GridSearchCV, the average ROC AUC score can reach 0.7203.

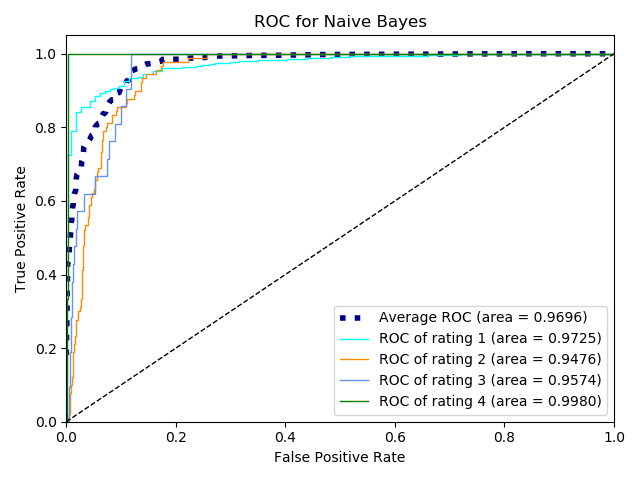
ROC chart as below:



* 1. Naive Bayes

We set multi\_class='one\_vs\_rest' and also use OneVsRestClassifier to improve the accuracy.

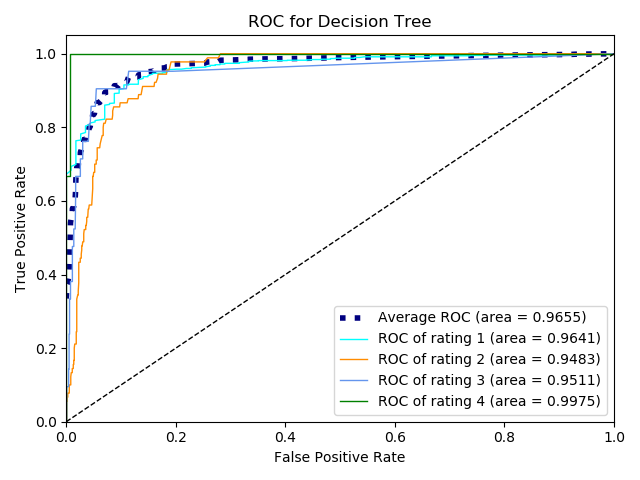
After optimized with GridSearchCV, the average ROC AUC score reaches 0.7118.

ROC chart as below:

* 1. Decision Tree

We use OneVsRestClassifier and BaggingClassifier to improve the accuracy. The final average ROC AUC score reaches 0.6879 after GridSearchCV optimization.

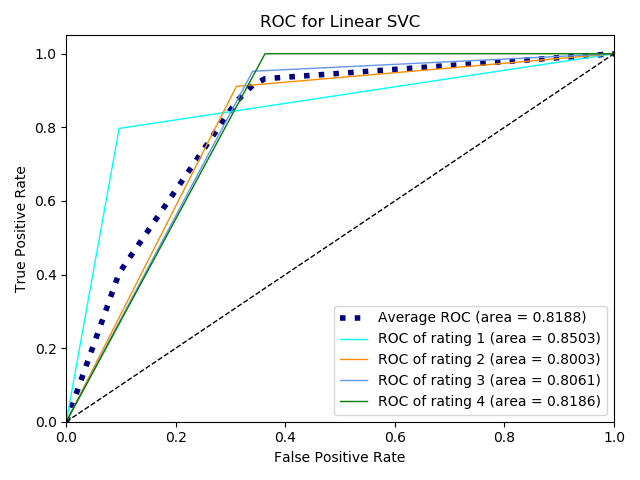
ROC chart as below:



* 1. Linear SVC

We also use OneVsRestClassifier and can get 0.8229 for average ROC AUC score.

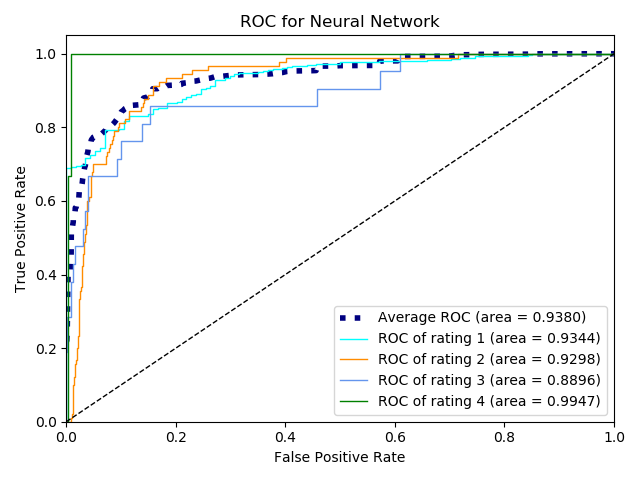
ROC chart as below:



* 1. Neural Network

We use OneVsRestClassifier and BaggingClassifier to improve the accuracy, and the average ROC AUC score is 0.7597 after GridSearchCV optimization.

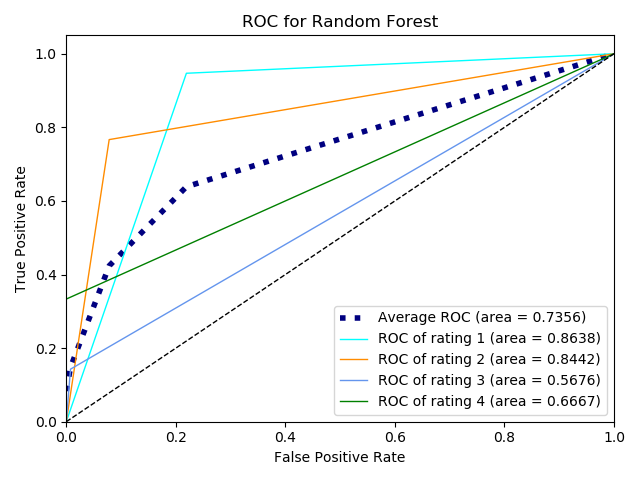
ROC chart as below:



* 1. Random Forest

We also add OneVsRestClassifier and the average ROC AUC score is 0.7988 after GridSearchCV optimization.

ROC chart as below:



* 1. XGBoost

We also use OneVsRestClassifier and the average ROC AUC score is 0.7988 after GridSearchCV optimization.

ROC chart as below:

* 1. HybridClassifier

Due to OneVsRestClassifier is not supported by sklearn.ensemble.VotingClassifier, so we have implemented a customized HybridClassifier. So it performs the voting with multiple classifier, and supports OneVsRestClassifier multiclass classification.

We select the best 3 classifiers: Linear SVC, Random Forest and XGBoost, and the average ROC AUC score is 0.7670.

1. Hyper-parameter Optimization
2. Post-Processing
3. Final Classification Model

# Project Performance & Validation

The project performance is measured as the time required for the validation process. The required time for processing 1188 records takes less than 1 second, so this is acceptable for the project requirement.

The average accuracy is 82.29% which is much better than the project success target 80%, so the project target achieves the project goal.

# Project Conclusions: Findings & Recommendation