2019

Predicting football match outcomes with machine learning algorithms

# 1 Abstract

Data science has finally permeated the realm of sports. Data Analytics in Football can provide teams, coaches and players with a firm grounding of performance analysis

The objective of this project is to explore different existing machine learning techniques to predict two football game outcomes: Full Time Goals and whether the game will be interesting or not (final goal difference is more or equal to 3 goals), by exploiting the results data of the Premier league in seasons 2016, 2017 and 2018..

To evaluate the model, Mean Squared Error (MSE) and explained variance score are used for linear regression, and covariance matrix is used for binary classification models.

Results showed about 66% of prediction accuracy from all models with original features. However, with data from the Principal Component Analysis, majority models showed decrease in accuracy except for K-Nearest Neighbors algorithm with 7% increase.

2 Introduction

Football is more of a business now than it has ever been. There is an unprecedented influx of money into football business and clubs are trying so hard to win the game, stay at the top of the league to ensure the popularity and the money comes from it. And this phenomenon has paved a way for data analytics to play an important part in football industry to increase the winning possibility.

The advance of new technology over the past years allowed us to collect new types (such as, player's running distance through GPS technology) and large amount of data per game. And the emergence of new Machine Learning techniques and tools made it possible to analyze those data with better predictive performance. Although it cannot have 100% accuracy, now we can develop better models that enable us to predict even the outcome of a match and the final score.

However, this does not mean an easy task. Unlike many other sports games, predicting results of football games can be very difficult due to the low-scoring nature of games. According to the research, the average goals per game has been less than 3 for many years, regardless of the league. (Planet Football 2017) Because of this limitation in data analytics, football clubs use data analytics mainly as a complementary tool to obtain a piece of additional information for the decision-making process, not the main source of information.

Therefore, this research aims to test different machine learning models on predicting outcomes of football matches, compare their performances and limitations, and finally seek to improve prediction accuracy by using different methods such as, Data imputation and Principal Component Analysis.

3 Data Analysis

3.1 Football Data Exploration

Total of 4 csv files, divided into 2 train files and 2 test files, are used for this football data science project.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 798 entries, 0 to 797
Data columns (total 13 columns):
HomeTeam     798 non-null int64
AwayTeam     798 non-null int64
HTHG         798 non-null int64
HTAG         798 non-null int64
HTR          798 non-null int64
HS           798 non-null int64
AS           798 non-null int64
HST          798 non-null int64
AST          798 non-null int64
HF           798 non-null int64
AF           798 non-null int64
HC           798 non-null int64
AC           798 non-null int64
dtypes: int64(13)
memory usage: 81.2 KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 798 entries, 0 to 797
Data columns (total 2 columns):
Interest     798 non-null int64
FTG          798 non-null int64
dtypes: int64(2)
memory usage: 12.6 KB
```

Figure 1. Train data information

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 342 entries, 0 to 341
Data columns (total 13 columns):
HomeTeam     342 non-null int64
AwayTeam     342 non-null int64
HTHG         342 non-null int64
HTAG         342 non-null int64
HTR          342 non-null int64
HS           342 non-null int64
AS           342 non-null int64
HST          342 non-null int64
AST          342 non-null int64
HF           342 non-null int64
AF           342 non-null int64
HC           342 non-null int64
AC           342 non-null int64
dtypes: int64(13)
memory usage: 34.9 KB
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 342 entries, 0 to 341
Data columns (total 2 columns):
Interest     342 non-null int64
FTG          342 non-null int64
dtypes: int64(2)
memory usage: 5.5 KB
```

Figure 2. Test data information

Each train file contains 798 observations and 13 columns as feature. While each test file contains 342 observation and 2 columns as features. Data type is all integers and it also can be checked that all files has no missing data by matching total number of data and the number of non-null data. Therefore, in this case, the data cleaning and imputation is unnecessary, and pre-processing is not required much.

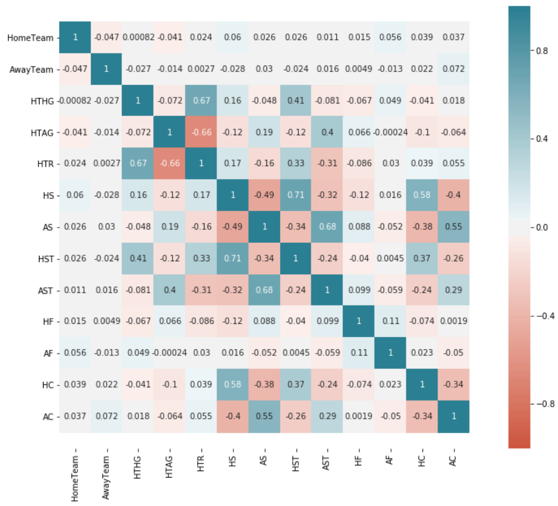


Figure 3. Correlation heatmap of football_train_x file

Based on the heatmap analysis, generated from the correlation matrix, some interesting pairs of features with high correlation are identified:

['HS', 'HST'] = 0.71, ['AS', 'AST'] = 0.68, ['HTHG', 'HTR'] = 0.67,
['HC', 'HS'] = 0.58, ['AC', 'AS'] = 0.55, ['HTHG', 'HST'] = 0.41

Correlation between some interesting pairs of features are highlighted by the pairplot in figure 4 and 5:

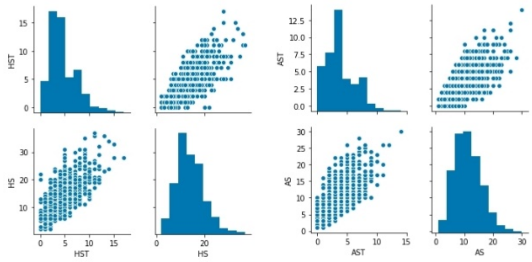


Figure 4. Visualization of correlation between 'Shots' and 'Shots on Target' by pairplot

As both Shots (HS [Home Shots], AS [Away Shots]) increase, both Shots on target (HST [Home Shots on Target] and AST [Away Shots on Target]) increase.

This relationship makes perfect sense because the more shots are made, the more shots on target can be made.

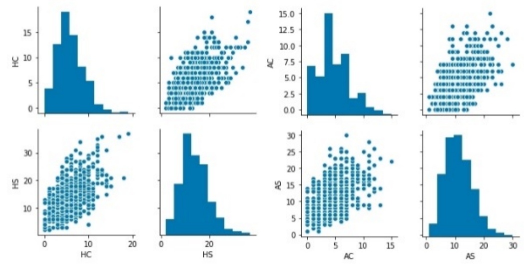


Figure 5. Visualization of correlation between 'Corner' and 'Shots' by pairplot

High correlation between 'Corner Kicks' and 'Shots' make sense for the similar reason. Corner kick is an important set piece in football and chances of scoring from corners are high as it often can be led to shots.

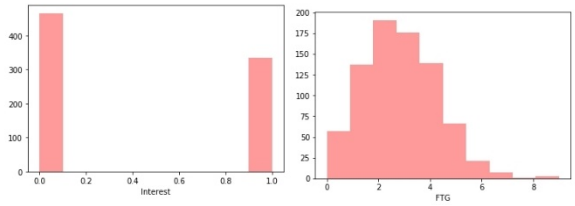


Figure 6. Histogram of target features

Figure 6 shows that different approach is required for two target features: 'Interest' and 'FTG'. Interest is a binary classification (0 and 1) while FTG is continuous.

In this research, linear regression will be used for continuous target variable, 'FTG', and logistic regression and KNN will be used for binary target variable, 'Interest'.

3.2 Principal Component Analysis

Principal Component Analysis (PCA) is used to reduces the dimensionality of the variable space by representing it with a few orthogonal (uncorrelated) variables that capture most of its variability. (ScienceDirect 2009) Since it is difficult to visualize high dimensional data, the PCA often can make data easy to explore and visualize.

First, among total of 13 features in the football_train_x file, two columns, 'HomeTeam' and 'AwayTeam', each indicates the index number of the team, are removed for the possible better PCA result. Therefore, total number of features is 11.

Then, to give more emphasis to those variables with higher variances while identifying the right principal components, the data is standardized by subtracting the mean, μ, from each value to be converted, and then divide the result by standard deviation, σ.

```
array([[ 1.63780053,  0.59791921,  1.22580442, ..., -0.23777807,
        -0.22000697, -0.19758706],
       [ 1.63780053, -0.71651476,  1.22580442, ...,  0.63262054,
        -0.22000697, -1.29001992],
       [-0.77803093, -0.71651476, -0.07845148, ..., -0.8180438 ,
        -1.86695496,  2.35142294],
       ...,
       [-0.77803093,  1.91235319, -1.38270739, ...,  0.63262054,
        -1.20817536, -0.56173135],
       [ 0.4298848 , -0.71651476,  1.22580442, ...,  1.79315202,
        -0.22000697, -0.19758706],
       [ 0.4298848 ,  0.59791921, -0.07845148, ...,  0.63262054,
         1.09755322, -0.92587564]])
```

```
array([[-0.82717683, -0.67252075, -0.17245937, ...,  0.04889961,
         2.44640358,  0.81745092],
       [-0.82717683,  0.65697114, -1.48315057, ...,  0.32305805,
         0.75067998, -0.66566959],
       [ 0.33699797,  0.65697114, -0.17245937, ...,  0.32305805,
        -0.26675418,  0.07589067],
       ...,
       [-0.82717683,  0.65697114, -1.48315057, ..., -0.77357574,
         0.41153526, -0.29488946],
       [-0.82717683, -0.67252075, -0.17245937, ...,  0.5972165 ,
        -1.28418834, -1.40722984],
       [ 1.50117276, -0.67252075,  1.13823184, ...,  1.41969185,
        -1.62333306,  3.04213169]])
```

Figure 7. Standardized football_train_x and football_test_x

After standardization, variables have a mean of 0 and a standard deviation of 1. Standardizing makes it easier to compare scores, even if those scores were measured on different scales. It also makes it easier to read results from regression analysis and ensures that all variables contribute to a scale when added together. (Statistics How To 2019)

Now it is a time for the PCA. The scaled data obtained from the standardization is used for this task.

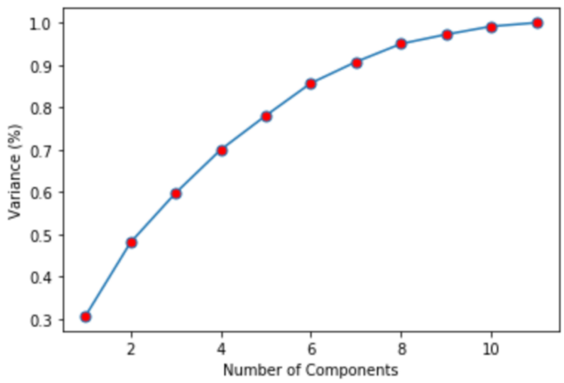


Figure 8. The plot of cumulative explained variance

Figure above shows that selecting 2 principal components allows us to preserve less than 50% of the total variance of the data which is not a good number as the rule of thumb is that the percentage of explained variance should be, at least, 60% (Hair, Black & Anderson 2014, 109).

Though 8 components seem optimal with almost 95% coverage of explained variance, for the simplicity of the analysis, 2 components will be used in this chapter, and both will be used and compared in the later modelling.

scaled_train_x.shape	scaled_test_x.shape
(798, 11)	(342, 11)

train_x_pca.shape	test_x_pca.shape
(798, 2)	(342, 2)

Figure 9. Change in the number of features after PCA

From the figure 9, it can be checked that the number of features is reduced from 11 to 2 principal components.

train_x_pca[:5]	test_x_pca[:5]
array([[1.94002488, 0.91920245], [1.57090415, 1.23135687], [-1.1605621 , 1.21064527], [-0.69116733, 0.40076752], [-0.56217411, -1.21275966]])	array([[1.38446853, -0.84501788], [-1.45127025, -1.91942893], [-0.73811313, -0.26277204], [3.43549761, 0.51315336], [-1.25242181, -1.49971542]])

Figure 10. Obtained values of 2 principal components

These values in Figure 10 will be used in further analysis such as, Linear Regression, Logistics Regression, and KNN analysis.

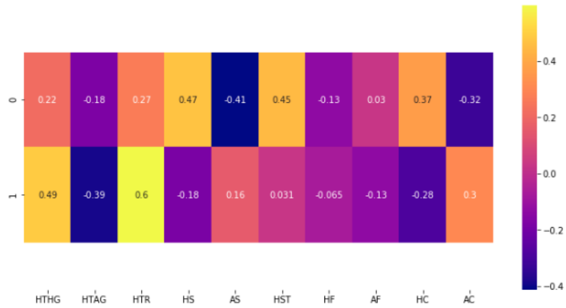


Figure 11. Contribution of each old feature towards new two principal components

From the heatmap above, it can be checked that the HTR ['Half Time Result'] made the biggest contribution to new components, followed by HTHG ['Half Time Home Goal'], HS ['Home Team Shots'] and HST ['Home Team Shots on Target']

When selecting 2 principal components, the second principal component should capture the highest variance from what is left after the first Principal Component explains the data as much as it can.

Since the PCA components are orthogonal to each other and they are not correlated, and it can be checked that features with high contribution on the first principal component has little contribution on the second principal component, and vice versa.

4 Methods

In this project, total of 3 machine learning algorithms will be used: Linear Regression for continuous target variable (FTG ['Full Time Goal']), and Logistic Regression and K-Nearest Neighbors algorithm for binary classification (Interest ['Interesting Match with more than or equal to 3 goals of difference'])

4.1 Linear Regression

Linear regression algorithm is widely used to quantify the correlation between attributes. In this chapter, the Linear Regression is used to predict the number of full-time goals.

First of all, train and test data should be set. For the train data, given 'football_train_x' and 'football_train_y', and for the test data, given 'football_test_x' and "football_train_y' are set. For the better possible prediction accuracy, the features, seemingly less correlated to the FTG, 'HomeTeam' and 'AwayTeam' are removed beforehand. Then, the FTG ['Full Time Goal'], in the files 'football_train_y' and 'football_test_y', are set to be the target value.

Then, independent variables (X) are standardized in order to reduce possible risk obtaining misleading results. The results with or without standardized data will be compared in the latter part of the report.

4.2 Logistic Regression

Logistic regression is similar to Linear regression in a form of predictive modelling technique which investigates the relationship between a dependent and independent variable.

However, Logistic regression is used when the dependent variable is binary in nature. In contrast, Linear regression is used when the dependent variable is continuous, and nature of the regression line is linear.

For this reason, logistic regression is chosen for classification task to predict whether the game was interesting or not.

Similar to the pre-processing done for the linear regression, the features 'HomeTeam' and 'AwayTeam' are dropped in order to increase the prediction accuracy, before performing logistic regression.

The result of the logistic regression can be checked by two methods: 'classification report' and 'confusion matrix' and they will be covered in more detail in the results chapter.

4.3 K-Nearest Neighbors

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. (Harrison, O. 2018)

In this project, KNN is used for the classification, whether the game will be interesting or not, as done with logistic regression in the previous chapter.

The KNN algorithm comes from the idea that similar things exist in close proximity. In other words, similar things are near to each other.

First, the optimal number of neighbors should be decided by checking the error rate of each K number.

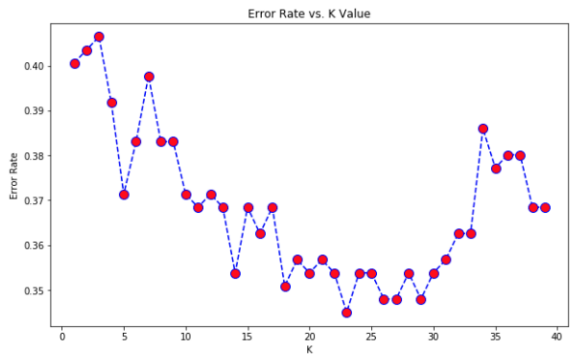


Figure 12. Error rate by different K values

According to the figure 12, it seems reasonable to choose the K value of 23 as it seems to have the lowest error rate. And the K value of 3 generates the highest error rate.

In this project, both cases, with K value of 3 and 23, will be covered and their results will be compared in detail later.

5 Experiments and Results

In this chapter, all results from 3 different machine learning algorithms will be compared and analyzed. Each algorithm is conducted with original features, standardized data, and PCA data with optimal number of principal components based on the analysis in chapter 3.2.

5.1 Linear Regression

	Coefficient
HTHG	0.865514
HTAG	0.738477
HTR	-0.095316
HS	-0.005946
AS	-0.013255
HST	0.164109
AST	0.183847
HF	-0.002153
AF	-0.024322
HC	-0.037650
AC	-0.012837

Figure 13. Correlation Coefficient of the Linear Regression

Regression coefficients are estimates of the unknown population parameters and describe the relationship between a predictor variable and the response variable. The value represents the mean change in the response given a one-unit change in the predictor. (Frost, J 2019)

For example, the first row in the figure 13 indicates that mean response value (FTG ['Full Time Goals']) increases by 0.865514 for every unit change in the predictor value (HTHG ['Half Time Home Goal'])

From the figure 13, it can be seen that HTHG and HTAG have the strongest positive correlation among 11 features.

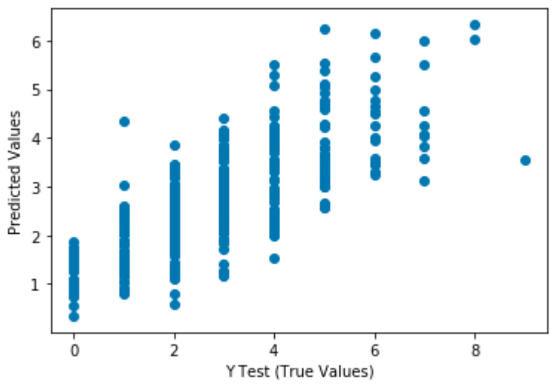


Figure 14. Correlation between predictions and true target values

From the figure 14, it can be seen visually that predicted values and true target values have positive correlation.

A residual is simply the vertical distance between a data point and the regression line, they are sometimes called "errors." In other words, itis the difference that isn't explained by the regression line. (Statistics How To 2019)

Each data point has one residual. They are positive if placed above the regression line and negative if placed below the regression line. If the regression line actually passes through the point, the residual at that point is zero. (Statistics How To 2019)

$$\text{Residual} = \text{Observed value} - \text{predicted value}$$
$$e = y - \hat{y}$$

Figure 15. Formula for calculating residual

And from the Figure 16, it can be visually checked whether the model was correct choice for the data or not.

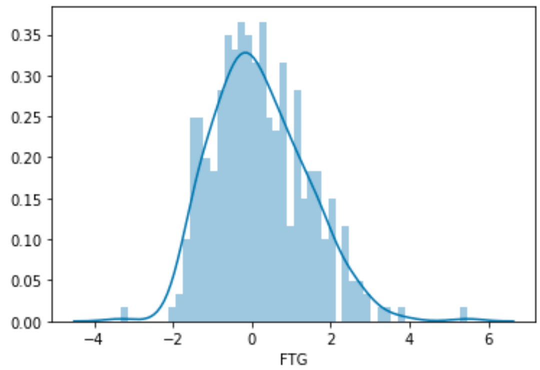


Figure 16. Histogram of the residual.

Not ideal distribution but the residuals can be considered well normally distributed, and therefore it can be said that our model was acceptable for the data.

Also, it is possible to evaluate the regression model with different metrics such as, Mean Absolute Error (MAE), Mean Squared Error (MSE) and Root Mean Square Error (RMSE), and the 'Explained Variance Score'.

In this project, MSE and Explained Variance Score are used as a metrics for evaluation of the prediction.

First, the mean squared error tells how close a regression line is to a set of points. It does by taking the distances from the points to the regression line (these distances are the "errors") and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It's called the mean squared error as finding the average of a set of errors. And the smaller MSE generally indicates a better estimate, at the data points in question. (Stephanie 2013)

The MSE is 1.4754325786712438 in this test.

Explained variance is used to measure the discrepancy between a model and actual data. In other words, it's the part of the model's total variance that is explained by factors that are actually present and isn't due to error variance. (Statistics How To 2019)

```
metrics.explained_variance_score(y_test, predictions)
0.5467572614215036
```

Figure 17. Explained Variance Score of Regression model

Best possible explained variance score is 1.0 and higher percentages of explained variance indicates a stronger strength of association, hence, better predictions.

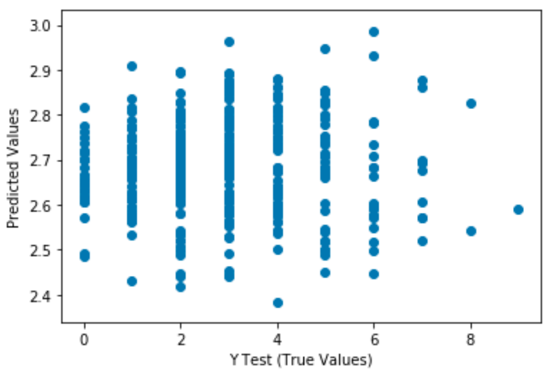
The results from the linear regression using principal components and standardized data are as following:

```
metrics.explained_variance_score(y_test, predictions)
0.5460720251435149
```

```
print("MAE:", metrics.mean_absolute_error(y_test, predictions))
print("MSE:", metrics.mean_squared_error(y_test, predictions))
print("RMSE:", np.sqrt(metrics.mean_squared_error(y_test, predictions)))
MAE: 0.9489268858420832
MSE: 1.4779766566720058
RMSE: 1.2157206326586738
```

Figure 18. Linear Regression result with standardized data

Both, Explained Variance Score and Mean Squared Error, were very close to those without standardization.

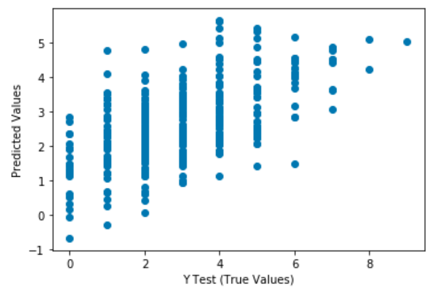



```
metrics.explained_variance_score(y_test, predictions)
0.0001369318615892512
```

Figure 19. Linear Regression result with 2 principal components

However, it is worth noting that the regression prediction was much better without PCA. Explained variance score dropped significantly and the MSE (Mean Squared Error) also increased to 3.1674096.

According to the Quora, unfortunately, PCA does not increase the accuracy of prediction. The main benefit to PCA is reducing the size of your feature vectors for computational efficiency. PCA is used to remove the least beneficial features so you have a smaller data set, but without losing too much predictive power. (Quora 2015)




```
metrics.explained_variance_score(y_test, predictions)
0.32838936132583896
```

```
print("MAE:", metrics.mean_absolute_error(y_test, predictions))
print("MSE:", metrics.mean_squared_error(y_test, predictions))
print("RMSE:", np.sqrt(metrics.mean_squared_error(y_test, predictions)))
MAE: 1.186794879930744
MSE: 2.151610366333091
RMSE: 1.4668368574361266
```

Figure 20. Linear Regression result with 8 principal components

With 8 principal components, which covers almost 95% of variance, explained variance score and MSE increased dramatically but still lower than that of without PCA.

In conclusion, Linear Regression didn't benefit from the PCA, regardless of the number of principal components, in terms of prediction accuracy.

5.2 Logistic Regression

```
              precision    recall  f1-score   support

           0       0.69      0.74      0.71       194
           1       0.62      0.56      0.59       148

    accuracy                           0.66       342
   macro avg       0.65      0.65      0.65       342
weighted avg       0.66      0.66      0.66       342
```

Figure 21. Classification Report on Logistic Regression

Figure above shows that the accuracy of logistic regression without PCA is 66% which is fairly good number.

This accuracy can be achieved from the confusion matrix, figure 22. A confusion matrix is a table that is often used to describe the performance of a classification on a set of test data for which the true values are known. (Data School 2014)

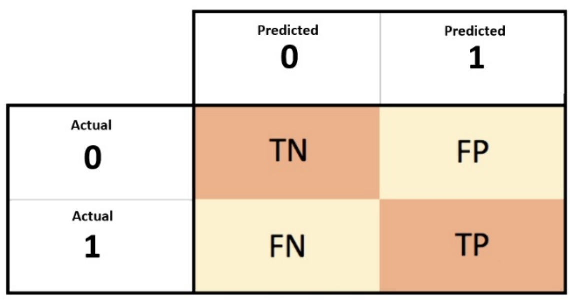


Figure 22. Components of the confusion matrix

One can calculate the accuracy rate by following formula:
(True Positive Num + True Negative Num) / Total Num

```
confusion_matrix(y_test, predictions)
```

```
array([[143,  51],
       [ 65,  83]], dtype=int64)
```

Figure 23. Confusion Matrix on Logistic Regression

By using the numbers from the figure 23 to the formula, our 66% accuracy can be calculated by (143 + 83) / 342.

The logistic regression result with PCA is as following:

```
print(classification_report(y_test, predictions))
```

```
              precision    recall  f1-score   support

           0       0.68      0.78      0.73       194
           1       0.64      0.51      0.57       148

    accuracy                           0.66       342
   macro avg       0.66      0.65      0.65       342
weighted avg       0.66      0.66      0.66       342
```

```
from sklearn.metrics import confusion_matrix
confusion_matrix(y_test, predictions)
```

```
array([[152,  42],
       [ 73,  75]], dtype=int64)
```

Figure 24. Regression results with 8 principal components

In the logistic regression algorithm, the prediction accuracy with 8 principal components, is same as without PCA.

Therefore, for this logistic regression, it can be said that the PCA allowed to conduct regression with smaller data without losing much predictive power. This implicitly tells that PCA lost virtually no information when it was reduced from 11 features to 8 features.

The reason why the PCA showed the better result with logistic regression than linear regression can probably be explained by the fact that logistic regression performs better with low dimensional data.

5.3 K-Nearest Neighbors

The k-nearest neighbors (KNN) algorithm is a simple, easy-to-implement supervised machine learning algorithm that can be used to solve both classification and regression problems. (Harrison, O. 2018)

In this project, KNN is used for the classification, whether the game will be interesting or not, as done with logistic regression in the previous chapter.

The KNN algorithm comes from the idea that similar things exist in close proximity. In other words, similar things are near to each other.

First, the optimal number of neighbors should be decided by checking the error rate of each K number.

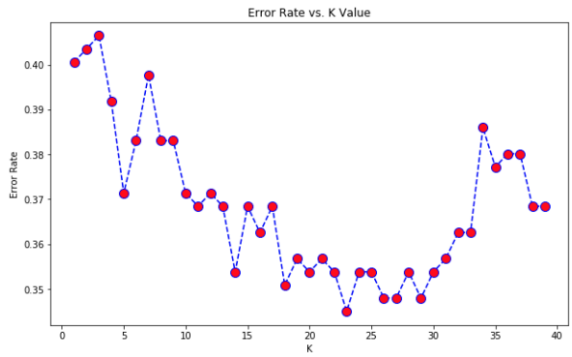


Figure 25. Error rate by different K values

According to the figure 25, it seems reasonable to choose the K value of 23 as it seems to have the lowest error rate. And the K value of 3 generates the highest error rate.

```
              precision    recall  f1-score   support

           0       0.65      0.87      0.74       194
           1       0.69      0.37      0.48       148

    accuracy                           0.65       342
   macro avg       0.67      0.62      0.61       342
weighted avg       0.66      0.65      0.63       342
```

```
print(confusion_matrix(y_test,pred))
```

```
[[169  25]
 [ 93  55]]
```

Figure 26. KNN results with K value of 23

With the K value of 23, the optimal choice with the least possible error rate, KNN algorithm showed 65% accuracy, very close to the number from the logistic regression in the previous chapter.

	precision	recall	f1-score	support
0	0.62	0.71	0.67	194
1	0.54	0.44	0.48	148
accuracy			0.59	342
macro avg	0.58	0.58	0.57	342
weighted avg	0.59	0.59	0.59	342

```
print(confusion_matrix(y_test,pred))
```

```
[[173  21]
 [ 74  74]]
```

Figure 27. KNN results with K value of 3

While the K value of 3, showed the highest error rate, ended up in 59% accuracy, 6% less than K value of 23.

Now the result with the PCA. By choosing 8 principal components with K value of 23, the prediction accuracy was increased by 7% as in figure 26 below.

	precision	recall	f1-score	support
0	0.70	0.89	0.78	194
1	0.78	0.50	0.61	148
accuracy			0.72	342
macro avg	0.74	0.70	0.70	342
weighted avg	0.73	0.72	0.71	342

```
print(confusion_matrix(y_test,pred))
```

```
[[173  21]
 [ 74  74]]
```

Figure 28. KNN results from 8 principal components (with K value of 23)

To sum up, unlike previous linear regression and logistic regression, the PCA helped boost prediction accuracy with K-Nearest Neighbors algorithm.

The maximum accuracy was 72%, obtained by choosing 23 as K-value and using 8 principal components. Without PCA, the best accuracy with 23 as K value generated 65% accuracy which is very close to logistic regression accuracy with or without PCA.

Among three machine learning techniques used in this project, KNN is the only algorithm that benefited from the PCA in prediction accuracy.

6 Conclusion and Discussion

Unfortunately, with given data, the prediction accuracy was rather disappointing with all methods. Although different usage, the best prediction model was KNN algorithm with 72% accuracy when done with 8 principal components.

For the continuous outcome, only the linear regression was conducted, and it showed poor results when combined with PCA. Without PCA, it showed fairly good prediction accuracy with 0.546 explained variance score and MSE of approximately 1.44 on the test data.

For the binary classification, obtained accuracies from both Logistic Regression and KNN, without PCA, were very similar. However, when PCA was implemented, KNN's accuracy was increased by 7% while Logistic Regression's accuracy remained almost same.

Predicting the outcome of sports game is a hard task due to uncertainties and numerous key factors, including injuries, weather and even luck. However, the prediction accuracy achieved from this simple research with simplified machine learning algorithms were fairly good and interesting.

Now the data science is used merely for a piece of additional data in the pro sports leagues, however, I believe, someday data science (AI) will be able to replace sports coaches or at least change the world of sports.

7 References

Planet Football 2017. Investigating which of Europe's four major leagues is most exciting. URL: www.planetfootball.com/quick-reads/investigating-europes-four-major-leagues-exciting/

Gonsales, A 2018. An Approach to Choosing the Number of Components in a Principal Component Analysis. URL: https://towardsdatascience.com/an-approach-to-choosing-the-number-of-components-in-a-principal-component-analysis-pca-3b9f3d6e73fe

ScienceDirect 2009. Comprehensive Chemometrics. URL: https://www.sciencedirect.com/topics/medicine-and-dentistry/principal-component-analysis

Statistics How To 2019. Standardized Variables: Definition, Examples. URL: https://www.statisticshowto.datasciencecentral.com/standardized-variables/

Frost, J 2019. Regression Coefficients. URL: https://statisticsbyjim.com/glossary/regression-coefficient/

Statistics How To 2019. Residual Values (Residuals) in Regression Analysis URL: https://www.statisticshowto.datasciencecentral.com/residual/

Statistics How To 2019. Explained Variance / Variation. URL: https://www.statisticshowto.datasciencecentral.com/explained-variance-variation/

Hair, J., Black, W. & Anderson, R. 2014. Multivariate Data Analysis URL: https://is.muni.cz/el/1423/podzim2017/PSY028/um/_Hair_-_Multivariate_data_analysis_7th_revised.pdf

Quora 2015. What is the motivation around a PCA? Is it computational or is it to increase accuracy? URL: https://www.quora.com/What-is-the-motivation-around-a-PCA-Is-it-computational-or-is-it-to-increase-accuracy

Data School 2014. Simple guide to confusion matrix terminology. URL: https://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/

Harrison, O. 2018. Machine Learning Basics with the K-Nearest Neighbors Algorithm. URL: https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761

Stephanie 2013. Mean Squared Error: Definition and Example URL: https://www.statisticshowto.datasciencecentral.com/mean-squared-error/