Predicting Football (Soccer) Player Performance metrics using Statistical Analysis and Machine Learning

Abstract:

In today's era, data stands as the new currency, being a part of every facet of our lives, from sports to agriculture and as well as robotics. Recognizing the transformative power of data, leading football teams like Leicester City, FC Barcelona, and the German national team are increasingly harnessing analytics to secure victories. In a remarkable instance, Leicester City strategically employed data analytics in 2014 to unearth the talent of French midfielder N'Golo Kante. The club's head of recruitment, Steve Walsh, leveraged data analytics to identify Kante's exceptional defensive prowess, having topped the charts for tackles and interceptions in Europe's premier leagues that season. Kante's subsequent signing played a pivotal role in Leicester City's unprecedented triumph.Esteemed teams now rely on it to evaluate player performance comprehensively, analysing metrics such as shots taken, passes completed, and tackles made. In our project, we aspire to leverage various football player features, employing machine learning to predict a player's performance. This initiative aligns with the evolving trend in sports, where data-driven insights not only enhance team decisionmaking but also contribute to the strategic edge that defines success on the field.

Introduction:

In this project, our primary objective is to predict a player's performance metric such as a player's playtime (minutes of playing) based on specific play metrics, and utilizing a dataset comprising 532 entries featuring a player's club, nationality, rank, age, passes attempted, penalty goals, and more. The playtime which we have considered as a performance metric, represents the minutes a player spends on the field in a single match.

Our methodology commenced with a meticulous examination of the dataset. Subsequently, we delved into exploratory data analysis, aiming to understand the nuanced interplay between various dataset features and a player's playtime. Employing correlation analysis, we pinpointed the features exerting substantial influence on playtime. To validate these findings, an analysis of variance was conducted to identify features with statistically significant impacts on playtime.

Further refinement involved the implementation of the Tukey HSD test for comparing mean minutes played among different player positions, revealing any noteworthy statistical differences. Following the comprehensive exploratory analysis, we constructed both linear regression and lasso regularization models. These models, leveraging metrics such as starts, matches played, minutes of play, and penalties, serve as predictive tools to estimate a player's playtime.

Background and Related work:

Previous research has explored various methods for predicting player performance in football. Linear regression has been widely used to model the relationship between player attributes, such as fitness levels, skill ratings, and historical performance statistics, and their on-field contributions.
Linear regression is a well-established technique in sports analytics for predicting various performance metrics. However, its limitations in handling multicollinearity and feature selection can be addressed by employing regularization techniques such as Lasso regression.

Below are some topics that will provide the background knowledge to understand the rest of this project.

Linear Regression- Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable.[1]

This form of analysis estimates the coefficients of the linear equation, involving one or more independent variables that best predict the value of the dependent variable. Linear regression fits a straight line or surface that minimizes the discrepancies between predicted and actual output values. There are simple linear regression calculators that use a “least squares” method to discover the bestfit line for a set of paired data.[1]

Lasso Regression- Lasso regression is a regularization technique. It is used over regression methods for a more accurate prediction. This model uses shrinkage. Shrinkage is where data values are shrunk towards a central point as the mean. The lasso procedure encourages simple, sparse models (i.e. models with fewer parameters). This particular type of regression is well-suited for models showing high levels of multicollinearity or when you want to automate certain parts of model selection, like variable selection/parameter elimination.[2]

Lasso Regression uses L1 regularization technique It is used when we have more features because it automatically performs feature selection.[2]

Below is some related work which we referred.

Work done in [3] outlines the use of data preprocessing techniques to reduce noise in the data and too and dimensionality reduction to identify the major attributes of the football players which affect the performance level more compared to certain attributes which barely affect their performance.

Work done in [4] showcases outcomes of different predictive models like Support vector machine, linear regression, K-means and random forest and then choosing the best model depending on parameters like F1 score, mean square error and precision.

Proposed Method:

1. Data extraction-

We used English Premier League 2020-21 data [5], this dataset has 532 unique rows and has the following features like Name, club, nationality, position, age, number of matches played, goals scored, penalty attempted etc. of a football player.

	Name	Club	Nationality	Position	Age	Matches	Starts	Mins	Goals	Assists	Passes_Attempted	Perc_Passes_Completed	Penalty_Goals	Penalty_Attempted	xG	xA	Yellow_Cards	Red_Cards
0	Mason Mount	Chelsea	ENG	MF,FW	21	36	32	2890	6	5	1881	82.3	1	1	0.21	0.24	2	0
1	Edouard Mendy	Chelsea	SEN	GK	28	31	31	2745	0	0	1007	84.6	0	0	0.00	0.00	2	0
2	Timo Werner	Chelsea	GER	FW	24	35	29	2602	6	8	826	77.2	0	0	0.41	0.21	2	0
3	Ben Chilwell	Chelsea	ENG	DF	23	27	27	2286	3	5	1806	78.6	0	0	0.10	0.11	3	0
4	Reece James	Chelsea	ENG	DF	20	32	25	2373	1	2	1987	85.0	0	0	0.06	0.12	3	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
527	Lys Mousset	Sheffield United	FRA	FW,MF	24	11	2	296	0	0	50	80.0	0	0	0.22	0.10	0	0
528	Jack O'Connell	Sheffield United	ENG	DF	26	2	2	180	0	0	77	77.9	0	0	0.00	0.00	0	0
529	Iliman Ndiaye	Sheffield United	FRA	MF	21	1	0	12	0	0	3	100.0	0	0	0.00	0.00	0	0
530	Antwoine Hackford	Sheffield United	ENG	DF,FW	16	1	0	11	0	0	1	100.0	0	0	1.16	0.00	0	0
531	Femi Seriki	Sheffield United	ENG	DF	17	1	0	1	0	0	0	-1.0	0	0	0.00	0.00	0	0

532 rows × 18 columns

Figure 1: Snapshot of how the dataset looks like

2. Exploratory data analysis-

We explored the dataset by finding out number of unique values, count of players in a club, minimum and maximum of all numerical columns, finding out if there is any relation between rest of the features and minutes of played by a player.

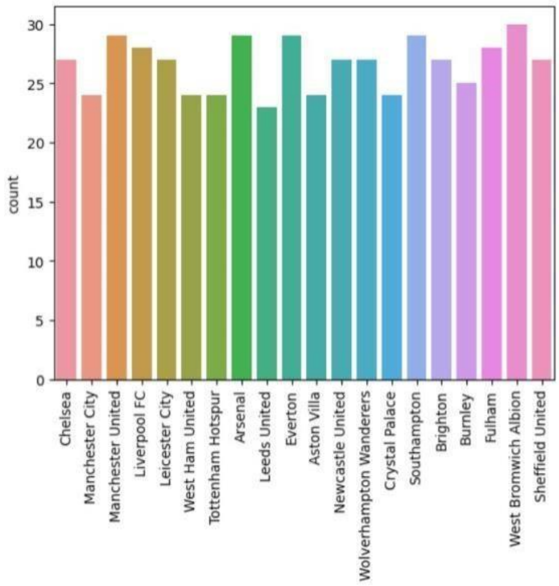

Figure 2: count of players in every club

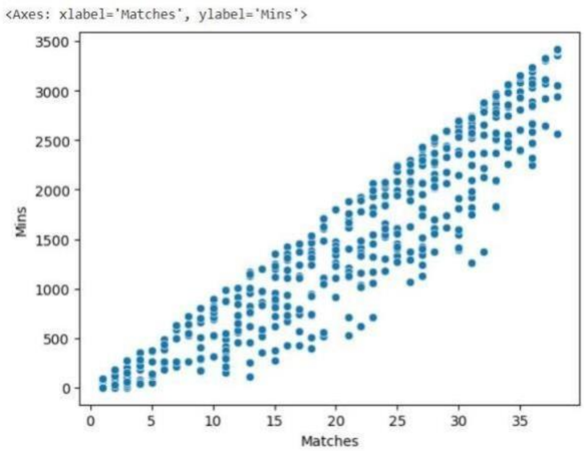

Figure 3: scatter plot depicting relation between matches and minutes played

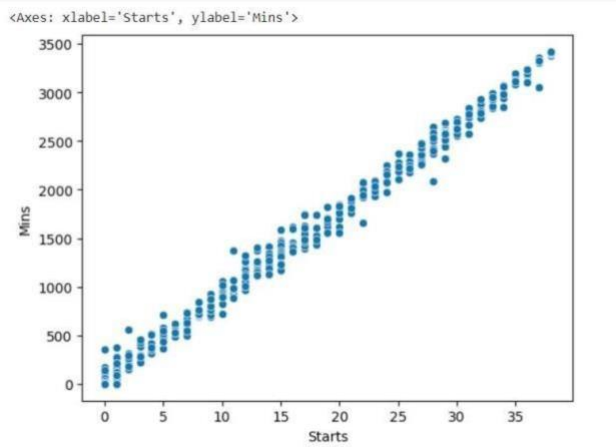

Figure 4: Scatter plot depicting relation between starts and minutes played

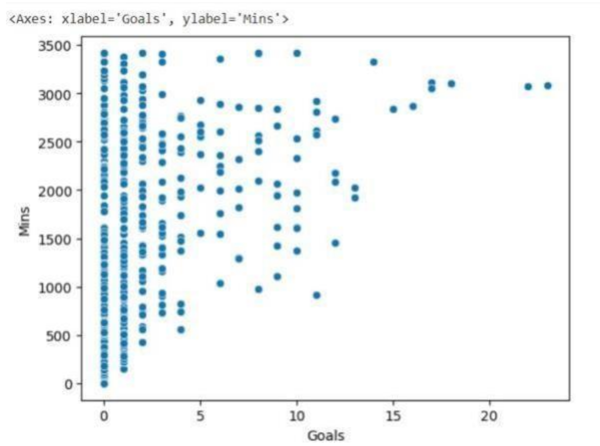

Figure 5: scatter plot depicting relation between goals and minutes played

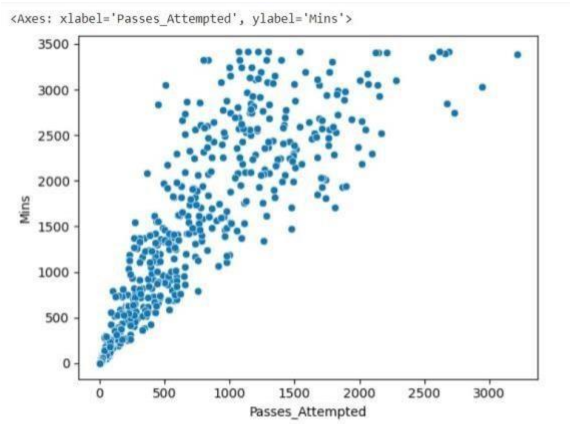

Figure 6: scatter plot depicting relation between passes attempted and minutes played

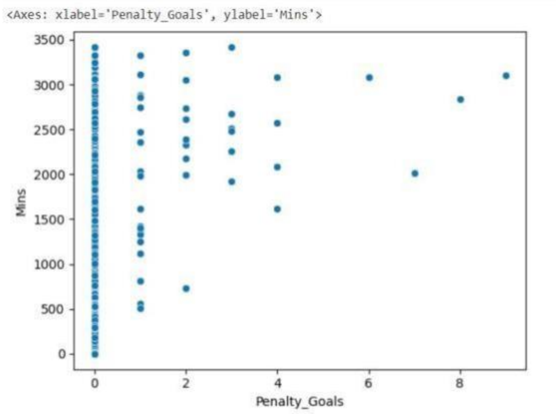

Figure 7: scatter plot depicting relation between penalty attempted and minutes played

3. Correlation analysis-

We performed correlation analysis on all features to figure out which features have greater impact on minutes of playing of a player.

```
correlation_with_mins

Mins                    1.000000
Starts                  0.997031
Matches                 0.947351
Passes_Attempted        0.855600
Yellow_Cards            0.604634
Assists                 0.474677
Goals                   0.400398
Penalty_Attempted       0.224115
Penalty_Goals           0.219027
Red_Cards               0.193038
Age                     0.158643
xA                      0.117602
Perc_Passes_Completed   0.085460
xG                      0.075016
Name: Mins, dtype: float64
```

Figure 8: correlation matrix for all features with minutes of play

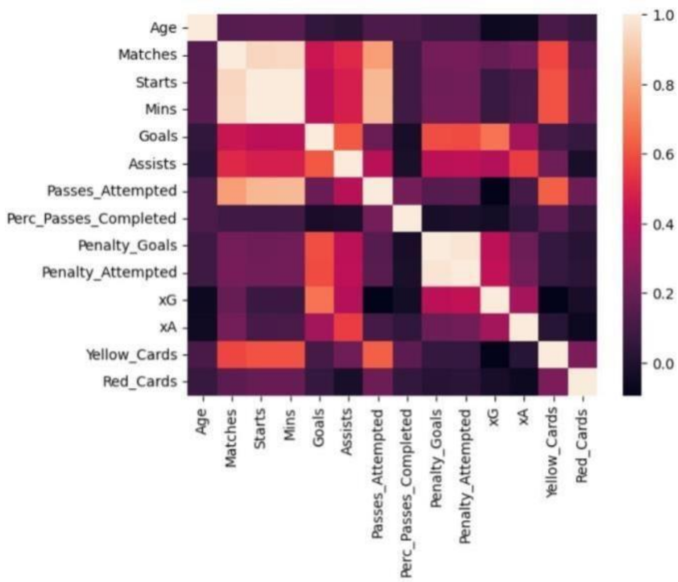

Figure 9: Heatmap for all features

From figures 8 and 9 we can conclude that Variables like 'Starts' and 'Matches' show a very strong correlation with 'Mins', which is expected as more starts or matches typically mean more playtime. Variables like 'Passes Attempted', 'Yellow Cards', 'Assists', and 'Goals' also show a significant correlation, suggesting that they might be good predictors of playtime.
This analysis can guide the feature selection for building a predictive model. However, correlation does not imply causation, and other factors (like the categorical variables Club, Nationality and Position, etc.) not captured in this data might also influence playtime.

4. Analysis of variance-

Analysis of variance is to test for differences among the means of the population by examining the amount of variation within each sample, relative to the amount of variation between the samples. We chose 'club,' 'nationality', and 'position' as our categorical variables.

```
[ ] anova_results

{'Club':                sum_sq     df         F    PR(>F)
 C(Club)    6.941538e+06   19.0  0.327653  0.997143
 Residual  5.708967e+08  512.0       NaN       NaN,
 'Nationality':                sum_sq     df        F    PR(>F)
 C(Nationality)  7.408342e+07   58.0  1.19932  0.159697
 Residual        5.037548e+08  473.0      NaN       NaN,
 'Position':                sum_sq     df         F    PR(>F)
 C(Position)  1.589256e+07    9.0  1.640316  0.100854
 Residual     5.619456e+08  522.0       NaN       NaN}
```

Figure 10: Analysis of variance results

Club:
F-value: 0.328
P-value: 0.997
Interpretation: The high P-value suggests that there is no statistically significant difference in the mean minutes played among different clubs.

Nationality:
F-value: 1.199
P-value: 0.160
Interpretation: The P-value indicates that there is no statistically significant difference in the mean minutes played among players of different nationalities.

Position:
F-value: 1.640
P-value: 0.101
Interpretation: The P-value is close to the typical alpha level of 0.05, suggesting that there might be some difference in the mean minutes played among different positions, but it's not statistically significant at the 0.05 level.

These results suggest that the categorical variables 'Club', 'Nationality', and 'Position' may not be strong predictors of a player's minutes played. However, for 'Position', the P-value is close to the typical alpha level of 0.05, suggesting that there might be some difference in the mean minutes played among different positions.

5. Tukey HSD test-

Tukey's Honest Significant Difference (HSD) test is a post hoc test commonly used to assess the significance of differences between pairs of group means. Tukey HSD is often a follow up to one-way ANOVA, when the F-test has revealed the existence of a significant difference between some of the tested groups. This test compares the mean minutes played between each pair of positions to identify any statistically significant differences.

Multiple Comparison of Means - Tukey HSD, FWER=0.05

group1	group2	meandiff	p-adj	lower	upper	reject
DF	DF,FW	-810.2846	0.6817	-2178.7015	558.1322	False
DF	DF,MF	99.982	1.0	-786.3943	986.3584	False
DF	FW	-212.8279	0.8795	-654.6958	229.0401	False
DF	FW,DF	-549.7846	0.9585	-1918.2015	818.6322	False
DF	FW,MF	-324.7244	0.6634	-865.3884	215.9396	False
DF	GK	137.882	0.9989	-427.669	703.4331	False
DF	MF	18.1228	1.0	-383.9978	420.2433	False
DF	MF,DF	75.1513	1.0	-872.0235	1022.326	False
DF	MF,FW	-324.5624	0.7886	-927.0398	277.915	False
DF,FW	DF,MF	910.2667	0.7248	-682.2483	2502.7816	False
DF,FW	FW	597.4568	0.9383	-797.4225	1992.3361	False
DF,FW	FW,DF	260.5	1.0	-1642.9194	2163.9194	False
DF,FW	FW,MF	485.5603	0.9865	-943.6907	1914.8113	False
DF,FW	GK	948.1667	0.5327	-490.6832	2387.0165	False
DF,FW	MF	828.4074	0.6667	-554.3948	2211.2096	False
DF,FW	MF,DF	885.4359	0.7785	-741.703	2512.5748	False
DF,FW	MF,FW	485.7222	0.988	-968.0384	1939.4828	False
DF,MF	FW	-312.8099	0.9871	-1239.5172	613.8975	False
DF,MF	FW,DF	-649.7667	0.9543	-2242.2816	942.7483	False
DF,MF	FW,MF	-424.7064	0.9329	-1402.3857	552.9729	False
DF,MF	GK	37.9	1.0	-953.7588	1029.5588	False
DF,MF	MF	-81.8593	1.0	-990.2866	826.568	False
DF,MF	MF,DF	-24.8308	1.0	-1274.1023	1224.4407	False
DF,MF	MF,FW	-424.5444	0.946	-1437.7168	588.6279	False
FW	FW,DF	-336.9568	0.999	-1731.8361	1057.9225	False
FW	FW,MF	-111.8965	0.9999	-716.4137	492.6207	False
FW	GK	350.7099	0.7491	-276.1645	977.5843	False
FW	MF	230.9506	0.8862	-253.6363	715.5375	False
FW	MF,DF	287.9791	0.9955	-697.0405	1272.9987	False
FW	MF,FW	-111.7346	0.9999	-772.1152	548.646	False
FW,DF	FW,MF	225.0603	1.0	-1204.1907	1654.3113	False
FW,DF	GK	687.6667	0.8844	-751.1832	2126.5165	False
FW,DF	MF	567.9074	0.9523	-814.8948	1950.7096	False
FW,DF	MF,DF	624.9359	0.969	-1002.203	2252.0748	False
FW,DF	MF,FW	225.2222	1.0	-1228.5384	1678.9828	False
FW,MF	GK	462.6064	0.5285	-237.4235	1162.6362	False
FW,MF	MF	342.8471	0.6753	-233.2559	918.9502	False
FW,MF	MF,DF	399.8756	0.9673	-633.2429	1432.9941	False
FW,MF	MF,FW	0.1619	1.0	-730.0249	730.3488	False
GK	MF	-119.7593	0.9998	-719.28	479.7615	False
GK	MF,DF	-62.7308	1.0	-1109.0884	983.6269	False
GK	MF,FW	-462.4444	0.626	-1211.2456	286.3568	False
MF	MF,DF	57.0285	1.0	-910.813	1024.87	False
MF	MF,FW	-342.6852	0.786	-977.1583	291.788	False
MF,DF	MF,FW	-399.7137	0.9736	-1466.4823	667.055	False

Figure 11: Tukey test results

Most comparisons between different positions do not show statistically significant differences in mean playtime (indicated by "False" in the 'reject' column and high p-values). The p-values for most position comparisons are well above the typical threshold of 0.05, suggesting that there's no strong evidence to reject the null hypothesis (which states that there's no significant difference in playtime across different positions). This analysis indicates that the player's position may not be a major determining factor in predicting their playtime.

Considering these findings, we focused on the numerical variables that showed stronger correlations with playtime for predictive modelling.

6. Model Building-

The linear regression model has been successfully built and evaluated. We utilized a train test split of 80-20. Here are the key findings:

```
mse, r2, coefficients
```

```
(4113.629119082633,
 0.9957361488518431,
 array([ 9.69989395e-01,  9.89984022e+00,  7.65879041e+01, -3.51224039e+00,
        -2.92729750e+00,  4.62107551e-02, -5.04575793e-01, -9.74953966e+00,
         2.06767703e+01, -2.83752333e+01, -3.68822265e+01, -2.03175568e+00,
        -3.90505744e+00]))
```

Figure 12: Mean square error and r2 score for linear regression

Mean Squared Error (MSE): The MSE of the model on the test set is approximately 4113.63. This value represents the average squared difference between the actual and predicted minutes played.

R-squared Value: The model has an R-squared value of approximately 0.996. This value indicates the proportion of variance in the dependent variable (minutes played) that is predictable from the independent variables. An R-squared value close to 1 suggests that the model explains a high portion of the variance. The high R-squared value suggests the model fits the data well.

Coefficients: The coefficients indicate how much the dependent variable is expected to increase (or decrease in the case of negative coefficients) when that independent variable increases by one, holding all other variables constant.

To further assess the model's performance and generalizability, we implemented cross-validation. This process involves dividing the dataset into multiple segments, using different segments as the training and test sets, and then averaging the results to get a more robust performance estimate. The results from the 5-fold cross-validation on your linear regression model are as follows:

```
[ ] cv_scores.mean()

    0.9952130749371669

[ ] cv_scores.std()

    0.00073912733373476
```

Figure 13: Cross validation results

Average R-squared Value: Approximately 0.995.
Standard Deviation of R-squared Values: Approximately 0.00074.
The high average R-squared value indicates that the model consistently explains a large portion of the variance in playtime across different subsets of the data. The low standard deviation suggests that the model's performance is stable across different folds of the data.
These results reinforce the model's robustness and its ability to generalize to unseen data.

7. Lasso Regularization

To refine the model, we implemented Lasso regression. Lasso regression, in particular, can shrink the coefficients of less important variables to zero, effectively performing feature selection. Here are the results:

```
mse_lasso, r2_lasso, coefficients_lasso

(4115.296193692407,
 0.9957344208987916,
 array([ 9.72685749e-01,  9.88237959e+00,  7.66085084e+01, -3.55502433e+00,
        -2.96548479e+00,  4.62121879e-02, -5.04899740e-01, -8.77533222e+00,
         1.98354396e+01, -2.73114647e+01, -3.49750985e+01, -2.02774648e+00,
        -3.77077465e+00]))
```

Figure 14: Lasso regression results

Mean Squared Error (MSE): The MSE for the Lasso Regression model is approximately 4115.30, which is very close to the MSEs from both the Linear and Ridge Regression models.
R-squared Value: The R-squared value is approximately 0.996, indicating a similar level of performance in explaining the variance in the target variable.

The alpha value used (0.01), Lasso Regression did not eliminate any of the variables (reduced coefficients to zero), which suggests that at this level of regularization, all variables are still considered relevant to the model. To further explore the potential for feature elimination, we experimented with higher alpha values. This increased the regularization strength, potentially leading to some coefficients being reduced to zero.

Adjusting the alpha value in the Lasso Regression model has shown varying effects on the coefficients of the variables. Here are the coefficients for different alpha values:

```
{0.05: array([ 9.83909314e-01,  9.81318381e+00,  7.66894684e+01, -3.72031377e+00,
       -3.11809898e+00,  4.62249447e-02, -5.06392201e-01, -5.16176920e+00,
        1.67087032e+01, -2.31118121e+01, -2.73975847e+01, -2.01108627e+00,
       -3.24051661e+00]),
 0.1: array([ 9.98378432e-01,  9.73146047e+00,  7.67846563e+01, -3.91689291e+00,
       -3.30782904e+00,  4.62562282e-02, -5.08525355e-01, -1.09131561e+00,
        1.31770957e+01, -1.79915878e+01, -1.80602704e+01, -1.99001031e+00,
       -2.58553494e+00]),
 0.5: array([ 1.05974764e+00,  9.52141773e+00,  7.69697717e+01, -4.33745206e+00,
       -3.55993167e+00,  4.69012246e-02, -5.19497973e-01,  0.00000000e+00,
        1.14729896e+01, -0.00000000e+00, -0.00000000e+00, -1.82062993e+00,
       -0.00000000e+00]),
 1: array([ 1.05086217e+00,  9.49778794e+00,  7.69301805e+01, -4.11637264e+00,
       -3.38794129e+00,  4.72587820e-02, -5.19775958e-01,  0.00000000e+00,
        1.05631000e+01, -0.00000000e+00, -0.00000000e+00, -1.63131437e+00,
       -0.00000000e+00]),
 5: array([ 9.80389127e-01,  9.28383405e+00,  7.66364277e+01, -2.34418895e+00,
       -2.00429833e+00,  5.00855339e-02, -5.22164328e-01,  0.00000000e+00,
        3.28088646e+00, -0.00000000e+00, -0.00000000e+00, -1.11859543e-01,
       -0.00000000e+00]),
 10: array([ 7.84679358e-01,  9.12556381e+00,  7.64205597e+01, -1.41576083e+00,
       -5.53145959e-01,  5.37708726e-02, -4.84093558e-01,  0.00000000e+00,
        0.00000000e+00, -0.00000000e+00, -0.00000000e+00, -0.00000000e+00,
       -0.00000000e+00])}
```

Figure 15: Adjusting alpha value in Lasso regression

Alpha = 0.05: Most coefficients are non-zero, indicating that most features are still contributing to the model.
Alpha = 0.1: Similar to alpha = 0.05, with slightly more shrinkage in coefficients.
Alpha = 0.5: Some coefficients are reduced to zero (e.g., Penalty Goals, xG, xA, Red Cards), indicating these features might be less important.
Alpha = 1: Similar pattern to alpha = 0.5, with further reduction in some coefficients.
Alpha= 5: Significant reduction in several coefficients, with many features now having zero coefficients.
Alpha = 10: Most features have coefficients reduced to zero, indicating a strong regularization effect.

It was found that as alpha increases, Lasso reduces the influence of less important features, and in some cases, coefficients are shrunk to zero, effectively eliminating those features from the model. The performance metrics for the Lasso Regression model with different alpha values are as follows:

```
{0.05: {'MSE': 4122.672720817001, 'R2': 0.9957267749947154},
 0.1: {'MSE': 4133.056637772416, 'R2': 0.9957160118765657},
 0.5: {'MSE': 4150.245259617182, 'R2': 0.9956981955584518},
 1: {'MSE': 4131.587459887319, 'R2': 0.9957175347060748},
 5: {'MSE': 4020.2715400523025, 'R2': 0.9958329156747662},
 10: {'MSE': 4011.8056598004414, 'R2': 0.9958416907131051}}
```

Figure 16: Lasso regression with different alpha values

Alpha= 0.05:
MSE: 4122.67
R2: 0.996

Alpha = 0.1:
MSE: 4133.06
R2: 0.996

Alpha = 0.5:
MSE: 4150.25
R2: 0.996

Alpha = 1:
MSE: 4131.59
R2: 0.996

Alpha = 5:
MSE: 4020.27
R2: 0.996

Alpha = 10:
MSE: 4011.81
R2: 0.996

Interestingly, as alpha increases, we see a slight improvement in the MSE and R2 scores, particularly for alpha values of 5 and 10. This suggests that simplifying the model by removing some of the less important features (as indicated by their coefficients being reduced to zero) does not significantly harm the model's performance. In fact, it slightly improved.
This improvement in performance with higher alpha values suggests that some of the features may not be contributing significantly to the model and that a simpler model might be more efficient and just as effective.
Exploring further, we delved deeper into the Lasso Regression models with higher alpha values (e.g., alpha = 5 and alpha = 10) where we saw improvements in performance.

The comparison of feature coefficients for the Lasso models with alpha values of 5 and 10 provides insights into which features are most influential. From below figure 13 we can conclude that we were successfully able to nullify some features which did have much impact on minutes played.

```
coefficients_comparison
```

	Feature	Coefficients (Alpha=5)	Coefficients (Alpha=10)
0	Age	0.980389	0.784679
1	Matches	9.283834	9.125564
2	Starts	76.636428	76.420560
3	Goals	-2.344189	-1.415761
4	Assists	-2.004298	-0.553146
5	Passes_Attempted	0.050086	0.053771
6	Perc_Passes_Completed	-0.522164	-0.484094
7	Penalty_Goals	0.000000	0.000000
8	Penalty_Attempted	3.280886	0.000000
9	xG	-0.000000	-0.000000
10	xA	-0.000000	-0.000000
11	Yellow_Cards	-0.111860	-0.000000
12	Red_Cards	-0.000000	-0.000000

Figure 17: Coefficients Comparison for alpha=5 and alpha=10

Results and Discussion:

We validated the reduced models by performing cross-validation on the models with alpha = 5 and alpha = 10 to check their robustness. The cross-validation results for the Lasso models with alpha values of 5 and 10 are as follows:

```
[ ] print(cv_scores_lasso_5.mean(), cv_scores_lasso_5.std())

    0.9953593383604034 0.0007145707848034078

[ ] print(cv_scores_lasso_10.mean(), cv_scores_lasso_10.std())

    0.9953429479534892 0.0007060261549322148
```

Figure 18: Cross validation results for lasso model with alpha values of 5 to 10

Alpha = 5: Average R-squared Value: Approximately 0.995.
Standard Deviation of R-squared Values: Approximately 0.00071.

Alpha = 10: Average R-squared Value: Approximately 0.995.
Standard Deviation of R-squared Values: Approximately 0.00071.

These results indicate that both models (with alpha = 5 and alpha = 10) perform consistently across different subsets of the data, as evidenced by the high average R-squared values and low standard deviations. This consistency suggests that the models are robust and not overfitting, despite the simplification achieved through the higher regularization (alpha) values.
The fact that the model performance remains strong even after some features are eliminated (coefficients reduced to zero) confirms that those features were not critical for predicting playtime. This finding supports the idea that a simpler model is just as effective, if not more so, than a more complex one.
Given these results, we finalized the model with alpha = 10 for our predictive tasks. It offers the benefits of being easier to interpret and potentially more generalizable, with fewer variables to consider.
The Lasso Regression model with an alpha value of 10 is a prudent choice based on several key considerations:
Simplicity and Efficiency: The Lasso model with alpha = 10 simplifies the prediction by reducing the number of features. Several coefficients are shrunk to zero, indicating that these variables do not significantly contribute to predicting the player's playtime. This reduction in features leads to a more streamlined and efficient model, which is easier to interpret and faster to run.

Strong Performance Metrics: The model demonstrates high predictive accuracy, as indicated by an R-squared value of approximately 0.995. This means the model explains about 99.5% of the variance in playtime, which is exceptionally high. Additionally, the low standard deviation in the cross- validation scores suggests that this performance is stable across different data subsets.

Robustness and Generalizability: The consistent performance across the cross-validation folds implies that the model is robust and likely to generalize well to new, unseen data. This is an important aspect, as it indicates the model is not just tailored to the specifics of the training data but can adapt to other similar datasets.

Reduced Risk of Overfitting: By penalizing the inclusion of less important features, Lasso with a higher alpha helps reduce the risk of overfitting. Overfitting occurs when a model is too complex, capturing noise in the training data as if it were a true signal. A simpler model, like the one with alpha = 10, is less prone to this issue.

Practical Interpretation: With fewer variables to consider, the model becomes more practical and easier to interpret. This can be particularly valuable in real-world scenarios where explain ability is key, such as in sports analytics, where coaches and team analysts may use the model's outputs to make decisions.

In summary, the Lasso Regression model with an alpha value of 10 strikes a balance between maintaining high predictive accuracy and ensuring model simplicity and interpretability. Its robustness and generalizability make it a solid choice for predicting a player's playtime in future seasons.

Conclusion:

This project has successfully demonstrated the potent application of statistical analysis and machine learning techniques in predicting football player performance metrics, specifically focusing on playtime. The exploratory data analysis, backed by rigorous statistical tests such as correlation analysis, ANOVA, and the Tukey HSD test, laid a strong foundation for understanding the intricate relationships within the dataset. The subsequent adoption of linear regression followed by Lasso regularization provided a holistic approach to modelling. Notably, the Lasso model with an alpha value of 10 emerged as the optimal choice, striking a delicate balance between model complexity and predictive accuracy. This model not only exhibited high predictive power, as indicated by an Rsquared value of approximately 0.995, but also demonstrated robustness and generalizability across various subsets of data, confirmed through cross-validation techniques. The reduction of feature complexity in this model, without compromising its predictive capacity, underscores the significance of feature selection in machine learning. This approach not only enhances model efficiency and interpretability but also aligns with the evolving trends in sports analytics where data-driven insights are increasingly shaping strategic decisions. Our findings echo the transformative potential of data analytics in sports, reinforcing the fact that accurate, streamlined models can significantly influence player assessment and team strategy in football.

References:

[1] About linear regression. IBM. (n.d.). https://www.ibm.com/topics/linear-regression#:~:text=Resources-,What%20is%20linear%20regression%3F,is%20called%20the%20independent%20variable.

[2] Team, G. L. (2023, May 30). A complete understanding of lasso regression. Great Learning Blog: Free Resources what Matters to shape your Career! https://www.mygreatlearning.com/blog/understanding-of-lasso-regression/#:~:text=Lasso%20regression%20is%20a%20regularization,i.e.%20models%20with%20fewer%20parameters.

[3] Player performance prediction in football game | IEEE conference ... (n.d.-b). https://ieeexplore.ieee.org/document/8474750/.

[4] Sports analytics: A comparison of machine learning ... - IEEE xplore. (n.d.-c). https://ieeexplore.ieee.org/document/9935852.

[5] Chaudhari, R. (2021, June 3). English premier League(2020-21). Kaggle. https://www.kaggle.com/datasets/rajatrc1705/english-premier-league202021.