STAT 847: Final Project

DUE: Friday April 19, 2024 by 11:59pm Eastern

The dataset used is Fifa23 Players dataset. Here is the link.

https://www.kaggle.com/datasets/sanjeetsinghnaik/fifa-23-players-dataset

1) **MUST BE INCLUDED** Describe and justify two different topics or approaches you might want to consider for this dataset and task. You don't have to use these tasks in the actual analysis.

Fifa23 Players dataset is chosen for this final project

1. Player Performance Prediction:

Description: This approach involves building predictive models to forecast player performance based on their features. The objective could be to predict metrics such as goals scored, assists provided, or overall player ratings in the game.

Justification:

Useful for gamers: Gamers often want to know which players are likely to perform well in upcoming matches or seasons within the game. Predictive models can provide insights into player performance, helping gamers make informed decisions when selecting their teams.

Player evaluation: Predictive models can assist in evaluating players' potential and performance trajectory. This information can be valuable for in-game transfers, team management, and scouting purposes within the FIFA game.

Data-driven insights: By analyzing player features such as age, position, skill ratings, and historical performance data, predictive models can uncover patterns and relationships that might not be immediately apparent. These insights can inform strategies for gameplay and team composition.

2. Player Clustering and Profile Analysis:

Description: This approach involves clustering players based on their features to identify groups with similar characteristics. Once clusters are formed, you can analyze the profiles of each cluster to understand player archetypes or playing styles.

Justification:

Player diversity: The FIFA game includes a wide range of players with diverse attributes and playing styles. Clustering analysis can help categorize players into meaningful groups, allowing gamers to understand the diversity of options available and tailor their team compositions accordingly.

Tactical insights: By identifying clusters of players with similar playing styles, gamers can develop strategies that leverage the strengths of specific player archetypes. For example, if a cluster comprises fast and agile forwards, a gamer might adopt a counter-attacking strategy that exploits their speed on the break.

Player comparisons: Clustering analysis enables comparisons between players within and across clusters, highlighting similarities and differences in their attributes and performance. This information can guide decisions regarding player selection, team formation, and tactical adjustments during gameplay.

2) **MUST BE INCLUDED** Give a ggpairs plot of what you think are the six most important variables. At least one must be categorical, and one continuous. Explain your choice of variables and the trends between them.

Here's a breakdown of the selected variables from Fifa23 Players dataset and why they were chosen:

- Overall Rating: This variable represents the overall skill level of a player in the game. It's arguably the most important variable as it encapsulates multiple skills and abilities of a player into a single metric.
- Potential Rating: This variable indicates the maximum level of skill a player can achieve in the game with proper development. It's crucial for gamers and managers alike as it informs decisions about player growth and future performance.
- Total Stats: This variable likely sums up various player attributes such as speed, shooting, passing, defending, etc. It provides a comprehensive overview of a player's capabilities and can be a significant factor in determining their overall effectiveness in the game.
- Value in Euro: The monetary value assigned to a player within the game. This variable is essential for understanding the economic impact of players within the game, influencing transfer decisions and team budgets.
- Age: Age can have a significant impact on player performance and potential. Younger players typically have higher growth potential, while older players may have more experience but declining physical attributes.
- International Reputation: This categorical variable represents the perceived reputation of a player on an international level within the game. It can influence transfer decisions, endorsement deals, and player ratings.

```
df <- read.csv("Fifa 23 Players Data.csv")

library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.3.3

## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

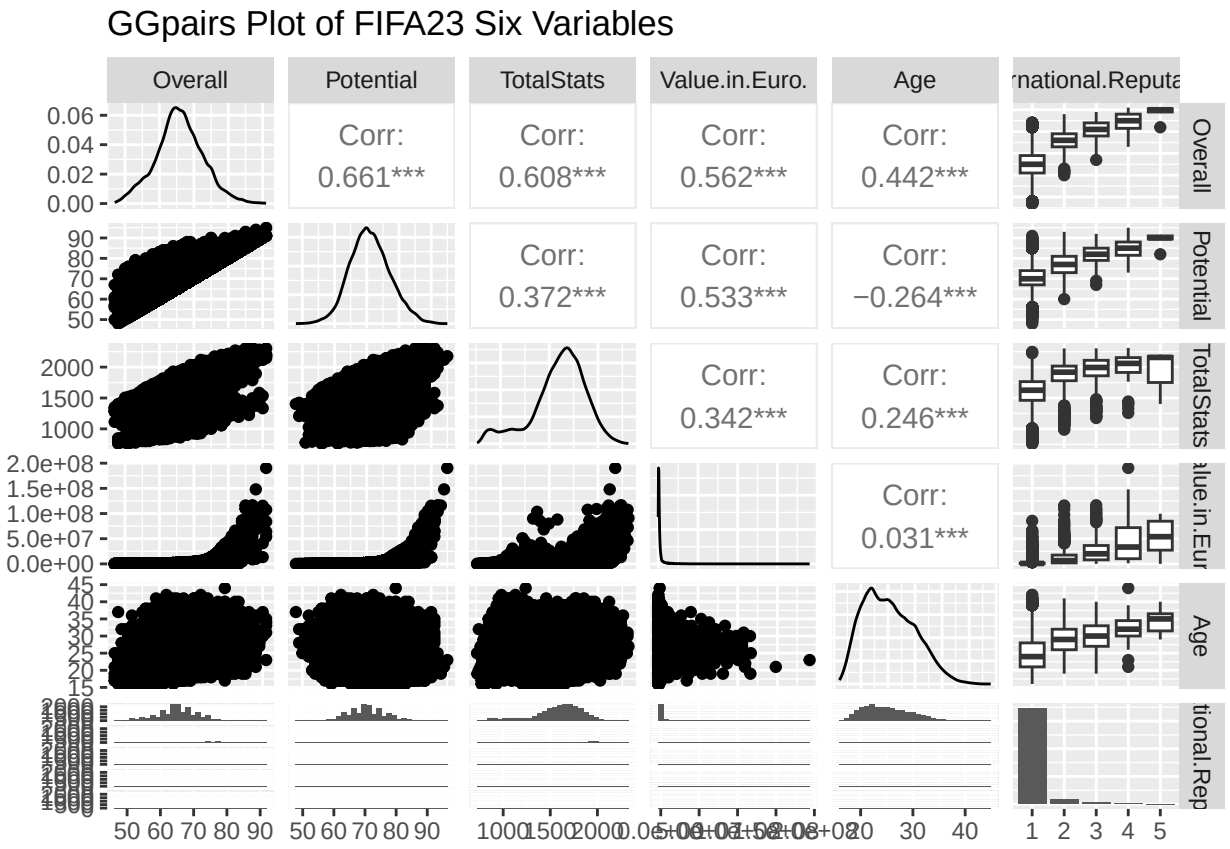
```
# Subset the dataset with selected variables
selected_vars <- c("Overall", "Potential", "TotalStats", "Value.in.Euro.",
                   "Age", "International.Reputation")
data_subset <- df[, selected_vars]

# Convert "International.Reputation" to a factor
data_subset$International.Reputation <- factor(data_subset$International.Reputation,
                                               levels = c(1, 2, 3, 4, 5),
                                               labels = c("1", "2", "3", "4", "5"))

# Plot ggpairs
ggpairs(data_subset, title = "GGpairs Plot of FIFA23 Six Variables")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



The ggpairs plot reveal several trends.

- Overall vs Potential:
  - There is a strong positive correlation (0.661) between a player's overall rating and their potential.
  - As a player's overall rating increases, their potential tends to increase as well.
- Overall vs TotalStats:
  - Another strong positive correlation (0.608) exists between overall rating and total stats.
  - Players with higher overall ratings tend to have higher total stats.
- Overall vs Value.in.Euro:
  - A moderate positive correlation (0.562) indicates that players with higher overall ratings also tend to have a higher value in euros.
- Potential vs TotalStats:
  - A moderate positive correlation (0.372) suggests that players with higher potential typically have higher total stats.
- Potential vs Value.in.Euro:

  - A strong positive correlation (0.533) shows that as a player's potential increases, their value in euros also tends to increase.

- TotalStats vs Value.in.Euro:
  - A moderate positive correlation (0.342) implies that players with higher total stats generally have a higher value in euros.

- Age:
  - Age has negative correlations with overall rating (-0.264), potential (-0.246), and total stats (-0.031).
  - As players age, these metrics tend to decrease.

- International Reputation:
  - If a player has higher overall, potential, totalstats, Value, he has a higher international reputation generally.

3) **MUST BE INCLUDED** Build a classification tree of one of the six variables from the last part as a function of the other five, and any other explanatory variables you think are necessary. Show code, explain reasoning, and show the tree as a simple (ugly) plot. Show the confusion matrix. Give two example predictions and follow them down the tree.

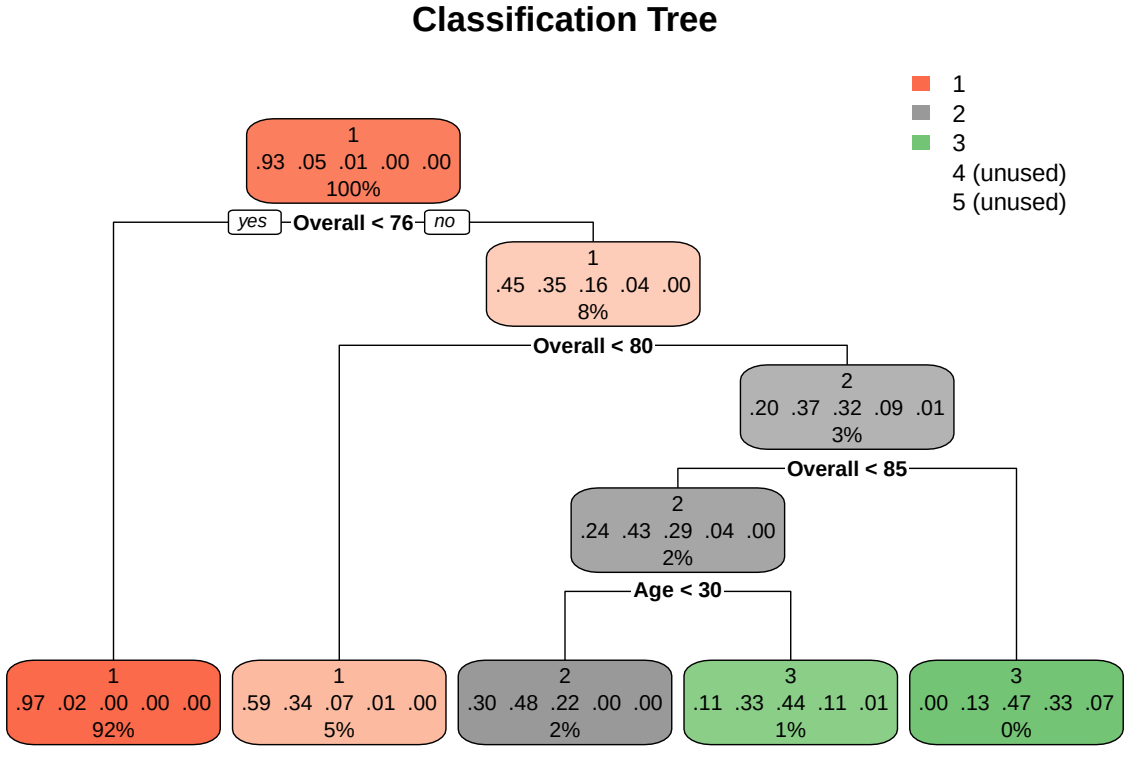
```
library(rpart)
library(rpart.plot)
```

```
## Warning: package 'rpart.plot' was built under R version 4.3.3
```

```
# Prepare the data
data <- df[, c("Overall", "Potential", "TotalStats", "Age",
               "International.Reputation", "Value.in.Euro.")]

# Build the classification tree
tree_model <- rpart(International.Reputation ~ Overall + Potential + TotalStats +
                      Age + Value.in.Euro. ,
                    data = data, method = "class", control = rpart.control(maxdepth = 5))

# Plot the tree
rpart.plot(tree_model, main = "Classification Tree")
```




```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.3.3
```

```r
predicted_classes <- predict(tree_model, type = "class")
conf_matrix <- table(data$International.Reputation, predicted_classes)
print(conf_matrix)
```

```
##    predicted_classes
##         1     2     3     4     5
##   1 17225    85    15     0     0
##   2   708   134    55     0     0
##   3    91    63   101     0     0
##   4    10     0    45     0     0
##   5     0     0     7     0     0
```

The decision tree is a model that predicts the 'International Reputation' based on five features: 'Overall', 'Potential', 'TotalStats', 'Age', and 'Value.in.Euro.'.

- The root node starts with the 'Overall' feature. If the 'Overall' score is less than 76, it predicts a reputation of '1'.
- For 'Overall' scores of 76 or higher, the tree splits further based on other features.
- A significant split occurs at 'Overall' score of 85 and 'Age' less than 30, leading to a prediction of '3', indicating a high reputation.
- The leaves of the tree represent the final prediction of the 'International Reputation' with varying probabilities.

```r
example1 <- data[1000, ]
# Following example 1 down the tree
pred1 <- predict(tree_model, example1, type = "class")
print(example1)
```

```
##      Overall Potential TotalStats Age International.Reputation Value.in.Euro.
## 1000      77        86       1896  20                         1       23000000
```

```r
print(pred1)
```

```
## 1000
##    1
## Levels: 1 2 3 4 5
```

For example 1, we have overall 77, so from root node, we go in the next node. Next node is Overall < 80 , which is yes, so we go to the leaf node with prediction "1". "1" is also the International Reputation of this player, so our tree worked good for this example.

```r
example2 <- data[50, ]
# Following example 2 down the tree
pred2 <- predict(tree_model, example2, type = "class")
print(example2)
```

```
##    Overall Potential TotalStats Age International.Reputation Value.in.Euro.
## 50      86        86       2108  28                         3       69500000
```

```r
print(pred2)
```

```
## 50
##  3
## Levels: 1 2 3 4 5
```

For example 2, we have Root node Overall < 76, but overall is 86, so we go to next node. Next node is Overall < 80, but w have overall 86. We go to the node Overall < 85. Then we go to leaf node with prediction "3" which is the real reputation.

4) **MUST BE INCLUDED** Build another model using one of the continuous variables from your six most important. This time use your model selection and dimension reduction tools, and include at least one non-linear term.

```
set.seed(123)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.3.3
```

```
## Loading required package: lattice
```

```
library(leaps)
```

```
## Warning: package 'leaps' was built under R version 4.3.3
```

```
library(MASS)

# Prepare the data
selected_vars <- c("Overall", "Potential", "TotalStats", "Age",
                   "International.Reputation", "Value.in.Euro.")
data <- df[, selected_vars]

# Remove any rows with missing values
data <- na.omit(data)

# Perform feature selection using stepwise regression
step_model <- stepAIC(lm(Value.in.Euro. ~ ., data = data), direction = "both")
```

```
## Start:  AIC=574113.7
## Value.in.Euro. ~ Overall + Potential + TotalStats + Age + International.Reputation
##
##                             Df  Sum of Sq        RSS    AIC
## - TotalStats                 1 3.1294e+13 5.2121e+17 574113
## <none>                                    5.2118e+17 574114
## - Potential                  1 1.3364e+15 5.2252e+17 574159
## - Age                        1 3.2979e+16 5.5416e+17 575249
## - Overall                    1 4.8476e+16 5.6966e+17 575761
## - International.Reputation   1 1.5409e+17 6.7527e+17 578914
##
## Step:  AIC=574112.8
## Value.in.Euro. ~ Overall + Potential + Age + International.Reputation
##
##                             Df  Sum of Sq        RSS    AIC
## <none>                                    5.2121e+17 574113
## + TotalStats                 1 3.1294e+13 5.2118e+17 574114
## - Potential                  1 1.3057e+15 5.2252e+17 574157
## - Age                        1 3.3258e+16 5.5447e+17 575258
## - Overall                    1 5.6338e+16 5.7755e+17 576014
## - International.Reputation   1 1.5408e+17 6.7529e+17 578912
```

```r
# Selected features from stepwise regression
selected_features <- names(step_model$coefficients[-1])

# Subset the data with selected features
data_selected <- data[, c(selected_features, "Value.in.Euro.")]


# Dimension Reduction (PCA)
# We'll use PCA to reduce dimensionality
preProc <- preProcess(data_selected[, -ncol(data_selected)], method = "pca")
data_pca <- predict(preProc, data_selected[, -ncol(data_selected)])

data_pca$Value.in.Euro. <- df[, "Value.in.Euro."]

# Non-linear Term (Polynomial Regression)
# We'll introduce polynomial terms for TotalStats
data_pca$PC1_sq <- data_pca$PC1^2

# Split data into training and testing sets
trainIndex <- createDataPartition(data_pca$Value.in.Euro., p = .8, list = FALSE)
train_data <- data_pca[trainIndex, ]
test_data <- data_pca[-trainIndex, ]

# Model Building
# Define formula with polynomial terms
formula <- as.formula("Value.in.Euro. ~ PC1 + PC2 + PC3 + PC1_sq")

# Build the model on the training data
model <- lm(formula, data = train_data)

# Summary of the model
summary(model)
```

```
##
## Call:
## lm(formula = formula, data = train_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -75744421  -1263481    -13319    988571 107702096
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    61727      45137   1.368    0.171
## PC1          -670628      37769 -17.756   <2e-16 ***
## PC2          1506866      31467  47.887   <2e-16 ***
## PC3         -4368147      76332 -57.225   <2e-16 ***
## PC1_sq       1383890      13999  98.855   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4258000 on 14828 degrees of freedom
## Multiple R-squared:  0.6969, Adjusted R-squared:  0.6968
```

```
## F-statistic:  8523 on 4 and 14828 DF,  p-value: < 2.2e-16
```

```r
# Make predictions on the testing set
predictions <- predict(model, newdata = test_data)

# Evaluate model performance on the testing set
rmse <- sqrt(mean((predictions - test_data$Value.in.Euro.)^2))
rmse
```

```
## [1] 4408662
```

These steps are taken to build the model:

- Data Preparation:
  - We start by selecting specific variables of interest from our dataset. These variables include "Overall," "Potential," "TotalStats," "Age," "International.Reputation," and "Value.in.Euro."
  - Any rows with missing values are removed to ensure data quality.
- Feature Selection using Stepwise Regression:
  - We perform stepwise regression to identify the most relevant features for predicting the "Value.in.Euro." (player value in euros).
  - The stepwise algorithm iteratively adds or removes features based on their impact on the model's performance.
- Dimension Reduction with Principal Component Analysis (PCA):
  - To reduce dimensionality, we apply PCA to the selected features. PCA transforms the original features into a set of uncorrelated principal components (PCs).
  - These PCs capture the most significant variation in the data.
  - We also retain the original "Value.in.Euro." column.
- Non-linear Term (Polynomial Regression):
  - We introduce a polynomial term for the "TotalStats" feature.
  - Specifically, we create a new feature called "PC1_sq" by squaring the first principal component (PC1).
- Model Building:
  - We define a regression formula that predicts "Value.in.Euro." based on the first three principal components (PC1, PC2, and PC3) and the squared PC1 term (PC1_sq).
  - The linear regression model is built using the training data.
- Model Summary:
  - We summarize the model's coefficients, including their significance levels.
  - This helps us understand the impact of each feature on the predicted value.

Here is the model summary:

- Coefficients:
  - Intercept: The estimated intercept is 61727. While not statistically significant (p-value = 0.171), it represents the baseline value when all other features are zero.
  - PC1 (First Principal Component): The coefficient for PC1 is -670628. It is highly significant (p-value < 2e-16) and negatively associated with the target value.

  - PC2 (Second Principal Component): The coefficient for PC2 is 1506866. It is highly significant (p-value < 2e-16) and positively associated with the target value.
  - PC3 (Third Principal Component): The coefficient for PC3 is -4386872. It is highly significant (p-value < 2e-16) and negatively associated with the target value.
  - PC1_sq (Squared PC1): The coefficient for the squared PC1 term is 1383990. It is highly significant (p-value < 2e-16) and positively associated with the target value.

- Model Fit:
  - The multiple R-squared value is 0.6969, indicating that approximately 69% of the variability in the "Value.in.Euro." can be explained by the model.
  - The adjusted R-squared value (0.6968) accounts for the number of predictors and adjusts the R-squared value accordingly.

- F-Statistic:
  - The F-statistic tests the overall significance of the model.
  - The F-statistic value is 8523, with a very low p-value (< 2.2e-16), indicating that the model as a whole is highly significant.

- Residual Standard Error:
  - The residual standard error (RSE) measures the average deviation of the predicted values from the actual values.

7) **OPTIONAL: PICK 2 OF 4** Discuss briefly the steps you would take to make sure your analysis is reproducible and easy to evaluate by others, even if the data is updated later.

Ensuring reproducibility and ease of evaluation for the analysis of the FIFA 23 players dataset involves following best practices in data science and documentation. Here are the steps I would take:

- Data Versioning:
  - Keep a record of the dataset version used for analysis. This can be achieved by storing the dataset file with a version number or using version control systems like Git to track changes.
- Documentation:
  - Provide clear documentation for each step of the analysis, including data preprocessing, feature selection, model building, and evaluation.
  - Use comments in the code to explain complex operations, assumptions, and decisions made during the analysis.
  - Create a README file that outlines the purpose of the analysis, describes the dataset, and provides instructions for running the code.
- Script Organization:
  - Organize analysis scripts into logical sections or modules, making it easy for others to navigate and understand the workflow.
  - Use consistent naming conventions for variables, functions, and files to enhance readability and maintainability.
- Parameterization:
  - Parameterize the analysis pipeline by defining key parameters at the beginning of the script or in a separate configuration file. This allows for easy modification of input parameters without editing the code.
- Package Management:
  - Clearly specify the versions of R packages used in the analysis. This can be achieved by including a sessionInfo() output or a DESCRIPTION file for R projects.
  - Utilize package management tools like renv or packrat to create isolated environments with specific package versions.
- Data Preprocessing:
  - Document all data preprocessing steps, including missing value imputation, feature scaling, encoding categorical variables, and any transformations applied to the data.
  - Store any preprocessed data separately from the raw dataset to facilitate reproducibility.
- Model Training and Evaluation:
  - Clearly outline the methodology used for model selection, hyperparameter tuning, and model evaluation.
  - Provide code to generate performance metrics, visualizations, and diagnostic plots to assess model performance.
  - Include cross-validation or bootstrapping techniques to estimate the stability of the models.
- Sensitivity Analysis:
  - Perform sensitivity analysis to assess the robustness of the results to changes in key parameters or assumptions.
  - Document the sensitivity analysis and its impact on the conclusions drawn from the analysis.
- Version Control:

- Use version control systems like Git to track changes to the code, data, and documentation.
- Create descriptive commit messages to explain the rationale behind each change.

By following these steps, the analysis of the FIFA 23 players dataset will be transparent, reproducible, and easy to evaluate by others, even if the data is updated later.

8) **OPTIONAL: PICK 2 OF 4** Discuss briefly any ethical concerns like residual disclosure that might arise from the use of your data set, possibly in combination with some additional data outside your dataset.

When working with datasets like FIFA 23, several ethical concerns, including residual disclosure, may arise, particularly when combined with additional data outside the dataset. Here are some considerations:

- Residual Disclosure:
  - Residual disclosure refers to the potential for identifying individuals' sensitive information from the residuals (errors) of a statistical model. In the context of the FIFA 23 dataset, residual disclosure could occur if the model predicts player values with high accuracy, leading to residual patterns that reveal private information about specific players.
  - This could pose ethical concerns if the revealed information includes sensitive attributes such as player performance, health status, or personal characteristics.
- Privacy Risks:
  - Aggregating FIFA 23 data with external datasets, such as player injury records or social media profiles, could increase the risk of privacy breaches. For example, linking player performance data with medical records might inadvertently disclose sensitive health information without proper consent or anonymization.
  - Analyzing player attributes alongside demographic or socioeconomic data could also lead to unintended discrimination or profiling based on characteristics such as nationality, race, or socioeconomic status.
- Fairness and Bias:
  - The analysis of FIFA 23 data may uncover biases or unfair practices within the sports industry, such as disparities in player valuation based on nationality, ethnicity, or other factors. It's essential to assess and address any biases in the data and models to ensure fair treatment and opportunities for all players.
  - Moreover, using external data sources to supplement FIFA 23 data could introduce additional biases or reinforce existing ones, leading to unfair outcomes or perpetuating stereotypes.
- Data Security:
  - Combining FIFA 23 data with external sources raises concerns about data security and confidentiality. Unauthorized access to sensitive player information could result in privacy breaches, identity theft, or other malicious activities.
  - Safeguarding data through encryption, access controls, and secure storage practices is crucial to protect the privacy and security of players and other individuals involved.
- Transparency and Informed Consent:
  - Transparency and informed consent are essential ethical principles when using FIFA 23 data, especially if the analysis involves sensitive or personally identifiable information. Players and stakeholders should be informed about how their data will be used, who will have access to it, and the potential implications of the analysis.
  - Providing clear explanations of data handling practices, anonymization techniques, and the purpose of the analysis promotes trust and accountability among stakeholders.

Overall, ethical considerations should guide the responsible use of FIFA 23 data to mitigate privacy risks, ensure fairness and transparency, and uphold the rights and dignity of individuals involved. It's essential to prioritize ethical principles and adhere to relevant laws, regulations, and industry standards to promote ethical data practices in sports analytics and research.