Comprehensive Statistical Analysis of NBA Players

Matthew Badal-Badalian

The following dataset was pulled using the NBA Stats API and uploaded to Kaggle by Justinas Cirtautas, a Data Scientist residing in London, England. It contains demographic, biographical and box score details about all NBA players under guarantee contracts, from the 1996 to the 2022 season. Please note that the original dataset contained an additional 52 rows with missing data that was ultimately cleaned before being uploaded to Kaggle. The new dataset is saved as all_seasons.csv, containing 12843 rows (excluding the headers for variable/column names) and 22 columns of information. The variables from the dataset are found as follows:

Variable	Description
See:	https://www.kaggle.com/datasets/justinas/nba-players-data
index	Data record number
player_name	Name of player
team_abbreviation	Abbreviation code of team player played for at end of season
age	Age of player
player_height	Height of player in cm
player_weight	Weight of player in kg
college	College attended by player
country	Country player was born in
draft_year	Year player was drafted
draft_round	Draft round player was picked
draft_number	Number player was picked within his draft round
gp	Number of games played through season
pts	Average number of points scored per game
reb	Average number of rebounds scored per game
ast	Average number of assists distributed per game
net_rating	Team point differential per 100 poss. with player on court
oreb_pct	% of available offensive rebounds grabbed by player on court
dreb_pct	% of available defensive rebounds grabbed by player on court
usg_pct	% of team plays used by player on court
ts_pct	Player's shooting metric based on free throws, 2 & 3 pt shots
ast_pct	% of teammate field goals player assisted while on court
season	NBA season data record belongs to

Below are formulas associated with the calculation of certain variables, according to the author who curated the dataset:

Variable	Formula
usg_pct	(FGA + Possession Ending FTA + TO)/POSS
ts_pct	PTS/(2(FGA + 0.44*FTA)

Reflecting on Data Analysis Themes for This Application

Relationship Between Physical Attributes and Performance Indicators

The first objective is to determine the relationship between players' physical attributes and their performance indicators (i.e. whether the number of rebounds grabbed increases for taller players, if the usage rate is impacted by age, etc). The physical variables used are age, player_height, and player_weight, while the performance variables include: pts, reb, ast, net_rating, oreb_pct, dreb_pct, usg_pct, ts_pct, and ast_pct.

The initial step involves extracting and inspecting all data rows within these specified columns to ensure no data is missing. A correlation coefficient of -1 signifies a perfect negative linear relationship, whereas 0 indicates no linear relationship, and 1 indicates a perfect positive linear relationship. Furthermore, a heat map is created to easily visualize the matrix.

The top three performance attributes that exhibit for strongest positive or negative correlation are then identified for each physical attribute. Nine scatter plots are then plotted to visualize the relationship between each physical-performance trait pair to understand the specific trend between them in further detail. Finally depending on the findings, the analysis may be refined by removing outliers associated with certain physical traits and repeating the process. This systematic approach provides insights into how players' physical attributes influence their on-court performance, aiding in player assessment and strategic decision-making for coaching staffs.

Examining How College Background Influences NBA Success

The second topic delves into examining how college background influences NBA success. The variables extracted for the analysis include player_name, college, gp, pts, reb, ast, usg_pct, and ts_pct. After ensuring that there is no missing data, career statistics are calculated for each player. First, the data is modified to additionally contain season_total_pts, season_total_pts, season_total_pts, where season_total_ denotes the original metric multiplied by the number of games played for the season. Afterwards, the dataset is grouped by player_name to create a new data frame, all_players_career_stats_DF, which contains player names (player_name), their college of origin (college), and NBA career statistics (total_gp, avg_pts, avg_reb, avg_ast, avg_usg_pct, and avg_ts_pct). An important distinction is that avg_pts, avg_reb and avg_ast are calculated on a per game basis whereas avg_usg_pct and avg_ts_pct are calculated on a per season basis.

Following this, all_players_career_stats_DF is further grouped by college, with an additional variable num_players, tracking the number of players drafted from each college. Rows associated with colleges meeting a specified threshold (i.e. 50 players) are filtered out and stored in a new data frame all_players_career_stats_above_threshold_DF.

From all_players_career_stats_above_threshold_DF, a new dictionary all_players_career_stats_summary_DICT is created. This contains five data frames, each listing colleges alongside summary statistics (min, mean, max, std, q25, median, and q75) for specific performance traits: avg_pts, avg_reb, avg_ast, avg_usg_pct, and avg_ts_pct. For example, the data frame avg_pts_summary_DF contains the columns college, min_avg_pts, mean_avg_pts, max_avg_pts, std_avg_pts, q25_avg_pts, median_avg_pts, and q25_avg_pts. Each data frame also includes a row representing summary statistics for all player career averages from all_players_career_stats_DF, as opposed to the summary statistics within individual colleges. Tables are then produced for each data frame to display the summary statistics.

Finally, histograms and box plots are generated for each of the five performance traits: avg_pts, avg_reb, avg_ast, avg_usg_pct, and avg_ts_pct—using data from all_players_career_stats_DF, to visually represent the distribution of NBA career statistics. These visualizations can also be produced for colleges of interest. These tables and plots facilitate the identification of colleges that consistently produce higher-quality NBA players, which could be invaluable during recruiting sessions. Additionally, further research could explore the variance in coaching styles across colleges of different quality levels.

Identifying NBA Player Performance Trends Through ggpairs Plot

The objective of this task is to identify trends in NBA player performance. Among the list of potential variables, the five continuous variables: usg_pct, pts, reb, ast, and age stand out. The usage rate, or usage percentage (usg_pct) is an important metric that denotes the percentage of team possessions utilized by a player while on the court, reflecting the team's reliance on the player to generate offence.

Points (pts), rebounds (reb), and assists (ast) are the primary statistics that draw the most attention from NBA fans. The number of points scored directly influences the outcome of a game, as the team with the higher score wins. It is also a direct measure of a player's scoring ability.

A rebound occurs when a player secures the basketball after any missed shot attempt, including free throws and field goals. It often reflects a player's positioning and athleticism. Grabbing offensive rebounds gives a team extra chances to score. Offensive rebounds provide additional scoring opportunities for the team, while defensive rebounds limit the opponent's chances and enable transition play for the defending team.

An assist refers to a pass that directly leads to a made basket by a teammate. Good ball movement and passing improve a team's field goal percentage by creating more open-shot opportunities. Assists often showcase a player's court vision, playmaking skills, and willingness to involve teammates.

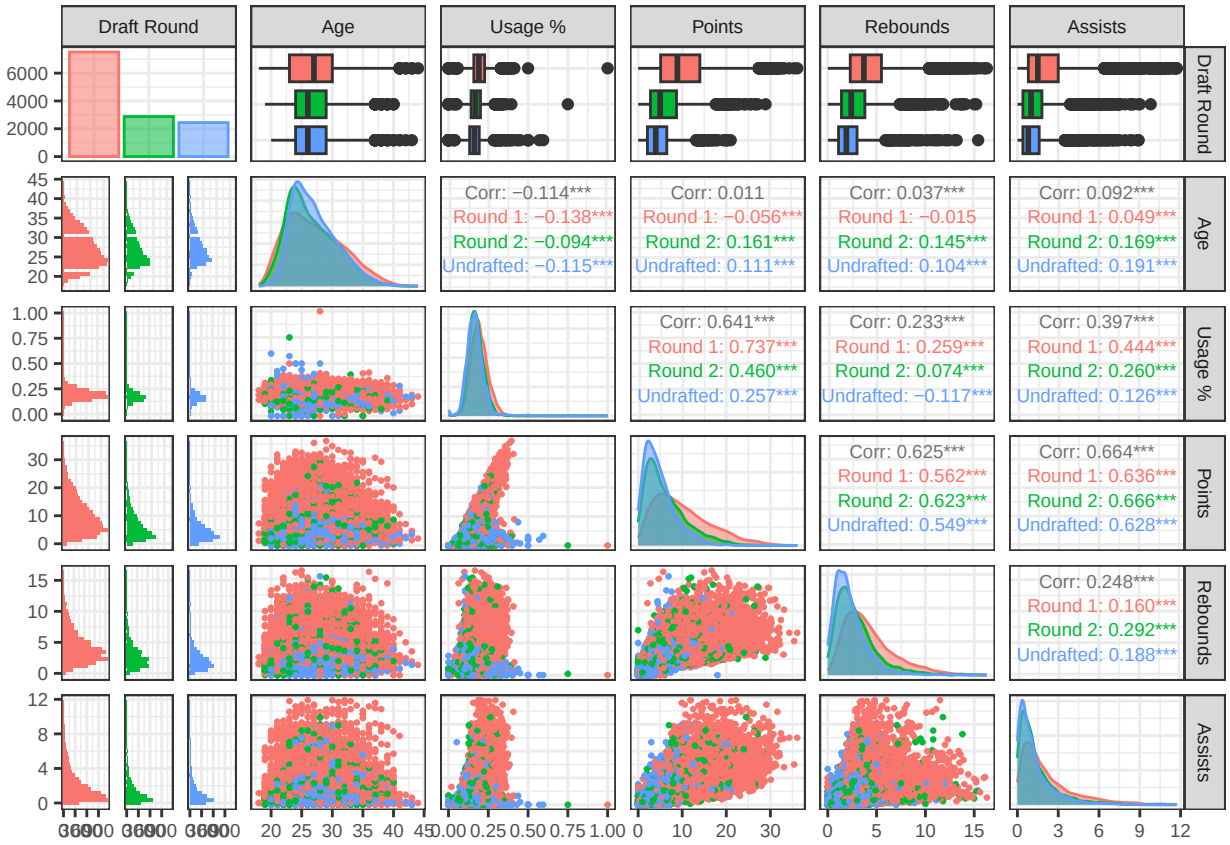
Age was selected as the final continuous variable because it can influence a player's performance level, durability and basketball intelligence. Younger players often exhibit greater growth potential, players in their prime typically deliver peak performance, and older players with experience excel at making strategic plays.

Finally, the round and position that a player is drafted can significantly influence their NBA career trajectory, opportunities and expectations. Players selected earlier in drafts are generally viewed as more impactful additions to their teams, being perceived as possessing greater skill and promise by NBA teams and scouts.

The draft_round and draft_number columns underwent analysis and transformation to create a new variable named modern_draft_round, which was selected as the categorical variable. This adjustment was made to make the analysis easier and align with the current NBA draft format, where picks 1 to 30 are designated as first-round players, picks 31 to 60 as second-round players and any remaining picks are categorized as undrafted players.

```
all_seasons_original_DF <- read.csv("all_seasons.csv")

all_seasons_modified_DF <- all_seasons_original_DF

pick_to_modern_round_number_DICT <- list(
  `Round 1` = as.character(1:30),
  `Round 2` = as.character(31:60)
)

ConvertToModernDraftRound <- function(draft_number) {
  for (modern_draft_round in names(pick_to_modern_round_number_DICT)) {
    if (draft_number %in% pick_to_modern_round_number_DICT[[modern_draft_round]]) {
      return(modern_draft_round)
    }
  }
  return("Undrafted")
}

all_seasons_modified_DF$modern_draft_round <-
  sapply(all_seasons_modified_DF$draft_number,ConvertToModernDraftRound)

write.csv(all_seasons_modified_DF,
          "Created_Data/all_seasons_modified.csv",row.names = FALSE)
```

```
library(GGally)

column_list <- c("modern_draft_round","age","usg_pct","pts","reb","ast")
ggpairs_plot <- ggpairs(all_seasons_modified_DF,
  mapping = aes(color = modern_draft_round),
  columns = column_list,
  columnLabels = c("Draft Round","Age","Usage %","Points","Rebounds","Assists"),
  lower = list(continuous = wrap("points",size = 0.5),
               mapping = aes(color = modern_draft_round)),
  upper = list(continuous = wrap("cor",size = 2.5)))

for (i in 1:length(column_list)) {
  ggpairs_plot[i, i] <- ggpairs_plot[i,i] +
  aes(color = modern_draft_round, alpha = 0.75)
}

new_ggpairs_plot <- ggpairs_plot + theme_bw() +
  theme(axis.text = element_text(size = 7),strip.text = element_text(size = 7))

new_ggpairs_plot
```



Several trends are revealed from the ggpairs plot. Firstly, the top left corner plot shows that the dataset includes over 7500 first-round players, nearly 3000 second-round players, and close to 2500 undrafted players, based on the modern NBA draft structure.

Examining the histograms, the distribution of categorical variable values (i.e. age) exhibits a similar shape

across first-round, second-round, and undrafted players, with slight variations in width and center. Predictably, first-round players show a higher frequency of values due to their larger representation in the dataset.

The slight changes in width and center from the histograms become more apparent through the boxplots. Median values are notably higher for first-round players and lower for undrafted players, with a decreasing interquartile range from first-round to undrafted players. Outliers are more prevalent among first-round players, aligning with the expectation of superior performance. However, the performance gap appears narrower between second-round and undrafted players. The increased age values among first-round players likely reflect longer NBA careers resulting from their success.

The diagonal density plots reveal a broader distribution of each given continuous variable for first-round players, suggesting a wider standard deviation. This makes sense, as most of the greatest NBA players were drafted in the first round, while some did not achieve significant success in their careers. Conversely, density curves for undrafted players have higher peaks, indicating greater concentration in the data around mean values.

One way to classify correlation value between two variables (typically noted as r) is to denote values of ± 0 to ± 0.2 as very weak, ± 0.2 to ± 0.4 as weak, ± 0.4 to ± 0.6 indicate as moderate, ± 0.6 to ± 0.8 as strong, and values of ± 0.8 to ± 1.0 as very strong. The sign of r determines whether the relationship between the two variables is positive or negative. A correlation of 0 signifies no relationship, while ± 1.0 shows a perfect linear relationship between the two variables.

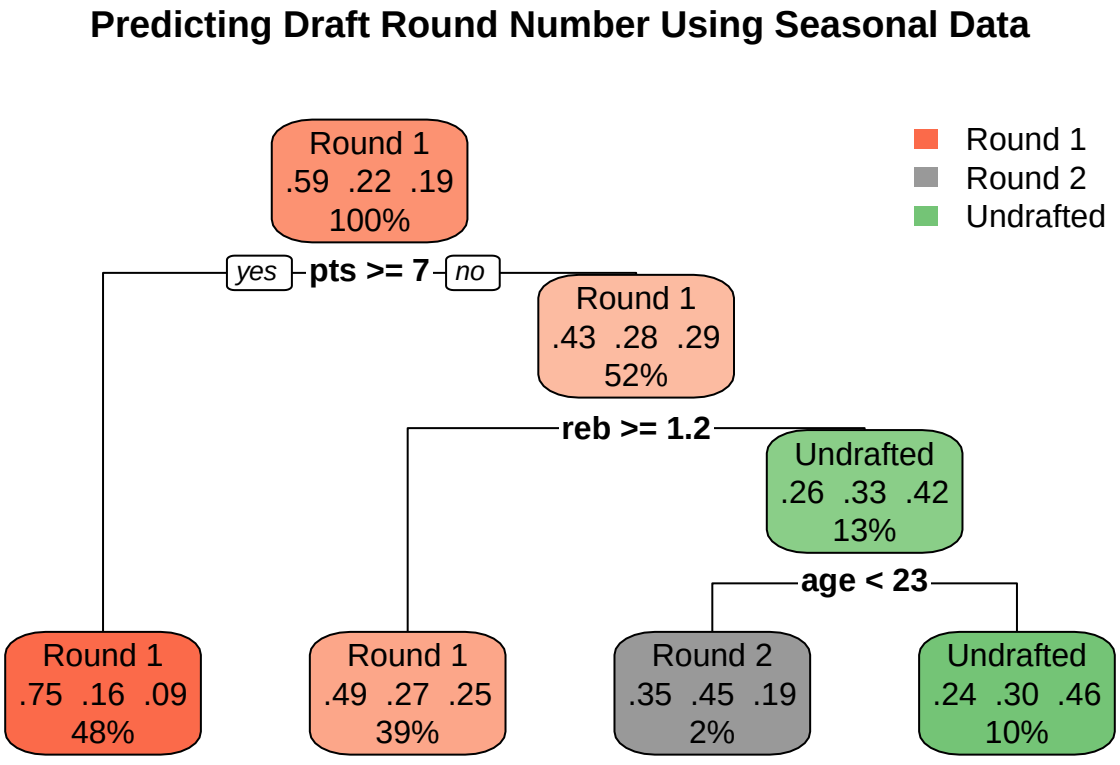
The relationship between age and other continuous attributes is observed as very weak, with correlation values falling mostly between -0.2 and 0.2. On the other hand, the usage rate exhibits a strong positive correlation with points scored by first-round players and a moderate positive correlation with points scored by second-round players and assists from first-round players. However, other combinations involving usage rates display weak or very weak correlations.

Moreover, there is a strong positive correlation between points scored and assists generated for all player types, and between points and rebounds for second-round players. In addition, a moderate positive correlation between points and rebounds for first-round and undrafted players. This is to be expected; players who score frequently tend to receive more playing time, which creates more opportunities to grab boards and generate more passes. Furthermore, there is a weak or very weak correlation between the number of rebounds and assists generated across all player types. Overall, these correlation observations align well with the scatter plot observations data.

Predicting NBA Draft Round Via Classification Trees Using Career vs. Seasonal Data

The modern_draft_round variable is selected as the target variable in this analysis, as it is the only categorical variable among the six variables considered in the previous question. When building a classification tree, additional seasonal statistics were incorporated as explanatory variables to assess potential improvements in accuracy. Surely enough, adding gp enhances the model's performance. This finding aligns intuitively with the notion that second-round and undrafted players typically receive fewer playing opportunities, particularly if their performance is subpar.

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

seasonal_data_column_list <- c("modern_draft_round","age","usg_pct","pts","reb","ast","gp")
seasonal_data_classification_tree_model <- rpart(modern_draft_round ~ ., method = "class",
                                    data = all_seasons_modified_DF[column_list])
rpart.plot(seasonal_data_classification_tree_model,
           main = "Predicting Draft Round Number Using Seasonal Data")
```



The observation point begins at the root of the classification tree and proceeds along a defined path, guided by decision criteria encountered at each branch split. Upon reaching a leaf node, the observation is classified into the majority class (or highest probable category. The class distribution is represented as the three decimal numbers within each node.

At the root node, these values describe the dataset's overall category distribution: 59% first-round, 22% second-round, and 19% undrafted. The percentages displayed at the bottom of nodes indicate the portion of observations which reach that node. When examining the distribution of classes within each node, this classification tree does not seem as accurate as desired. Notably, the model only assigns 2% of observations

to the second-round class and 10% to the undrafted class, significantly diverging from the dataset's actual class distribution.

```
seasonal_data_classification_tree_model_predictions <-
  predict(seasonal_data_classification_tree_model,
  all_seasons_modified_DF[seasonal_data_column_list],type = "class")

seasonal_data_confusion_matrix <-
  table(Prediction = seasonal_data_classification_tree_model_predictions,
  Truth = all_seasons_modified_DF$modern_draft_round)

row_labels <- rownames(seasonal_data_confusion_matrix)
updated_row_labels <- paste(row_labels,"(Predictions)",sep = " ")
column_labels <- colnames(seasonal_data_confusion_matrix)
updated_column_labels <- paste(column_labels,"(Truth Values)",sep = " ")
rownames(seasonal_data_confusion_matrix) <- updated_row_labels
colnames(seasonal_data_confusion_matrix) <- updated_column_labels

knitr::kable(seasonal_data_confusion_matrix,
    caption = "Confusion Matrix of Seasonal Data Classification Tree Model")
```

Table 3: Confusion Matrix of Seasonal Data Classification Tree Model

	Round 1 (Truth Values)	Round 2 (Truth Values)	Undrafted (Truth Values)
Round 1 (Predictions)	7104	2351	1773
Round 2 (Predictions)	98	126	54
Undrafted (Predictions)	321	400	617

```
CalculateAccuracies <- function(confusion_matrix,cat_labels) {
  accuracies_list <- numeric(length = nrow(confusion_matrix)+1)

  for (i in 1:nrow(confusion_matrix)) {
    true_value <- confusion_matrix[i,i]
    total_value <- sum(confusion_matrix[i,])
    accuracies_list[i] <- true_value/total_value
  }
  num_correct_predictions <- sum(diag(confusion_matrix))
  num_total_predictions <- sum(confusion_matrix)
  accuracies_list[nrow(confusion_matrix)+1] <-
    num_correct_predictions/num_total_predictions

  cat_accuracy_labels <- c(cat_labels,"Overall")

  accuracies_DF <- data.frame(accuracies_list)
  row.names(accuracies_DF) <- cat_accuracy_labels
  colnames(accuracies_DF) <- "Accuracy"

  return(accuracies_DF)
}

cat_labels <- c("Round 1","Round 2","Undrafted")
seasonal_data_accuracies_DF <-
```

```
  CalculateAccuracies(seasonal_data_confusion_matrix,cat_labels)

knitr::kable(seasonal_data_accuracies_DF,caption =
  "Seasonal Data Classification Tree Model Accuracies")
```

Table 4: Seasonal Data Classification Tree Model Accuracies

	Accuracy
Round 1	0.6327040
Round 2	0.4532374
Undrafted	0.4611360
Overall	0.6109467

The confusion matrix and accuracy table seem to confirm the suspicion, revealing that the model only correctly predicts approximately 45.32% of second-round data and 46.11% of undrafted data. It is important to note that the overall accuracy can be misleading, as it represents the ratio of correct predictions to total predictions without accounting for dataset distribution.

As an attempt to improve the model accuracy, the dataset was transformed to hold the career averages of each NBA player. The variables of the transformed data all_career_stats_DF can be explained below:

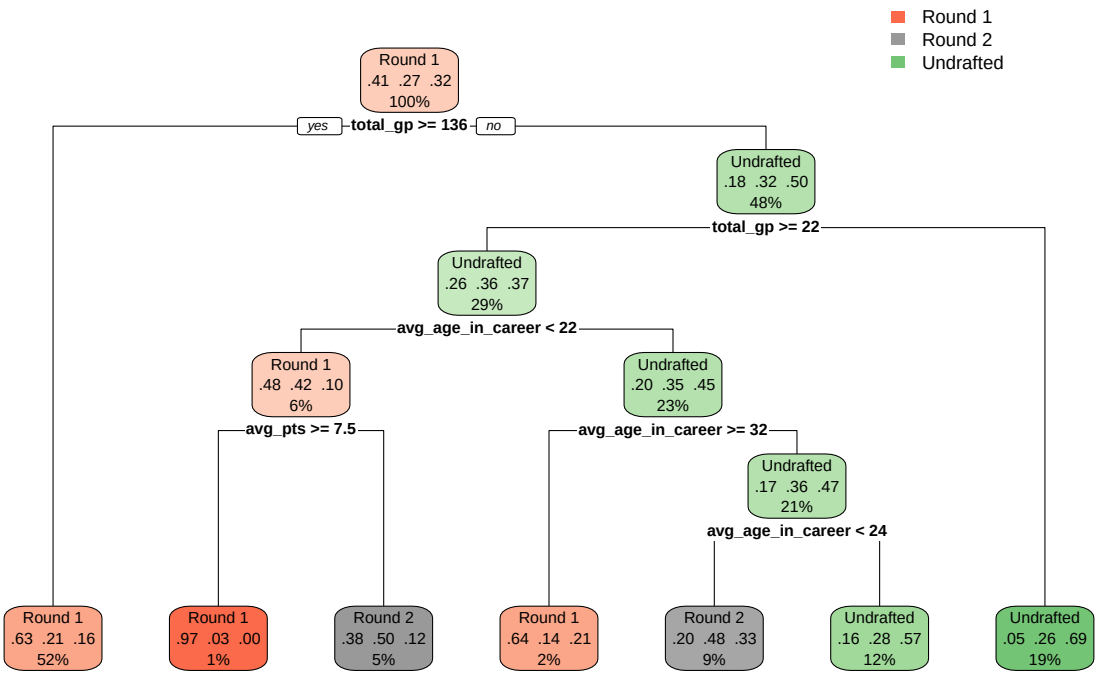
Variable	Description
player_name	Name of player
avg_age_in_career	Average age of player through their career
avg_player_height	Average height of player their career in cm
avg_player_weight	Average weight of player their career in kg
college	College attended by player
country	Country player was born in
draft_year	Year player was drafted
draft_round	Draft round player was picked
draft_number	Number player was picked within his draft round
total_gp	Number of games played through career
avg_pts	Average number of points scored per game (career)
avg_reb	Average number of rebounds scored per game (career)
avg_ast	Average number of assists distributed per game (career)
avg_net_rating	Team point differential per 100 poss. with player (career)
avg_oreb_pct	% of available offensive rebounds grabbed by player (career)
avg_dreb_pct	% of available defensive rebounds grabbed by player (career)
avg_usg_pct	% of team plays used by player on court (career)
avg_ts_pct	Player's shooting metric (career)
avg_ast_pct	% of teammate field goals player assisted (career)
modern_draft_round	Draft round player was picked based on modern format

```
library(plyr)
library(dplyr)

temp_all_seasons_modified_DF <- all_seasons_modified_DF
temp_all_seasons_modified_DF$season_total_pts <-
  temp_all_seasons_modified_DF$pts*temp_all_seasons_modified_DF$gp
temp_all_seasons_modified_DF$season_total_reb <-
  temp_all_seasons_modified_DF$reb*temp_all_seasons_modified_DF$gp
```

```
temp_all_seasons_modified_DF$season_total_ast <-
  temp_all_seasons_modified_DF$ast*temp_all_seasons_modified_DF$gp

temp_all_seasons_modified_GROUPED_DF <- group_by(temp_all_seasons_modified_DF,player_name)

all_career_stats_DF <- ddply(temp_all_seasons_modified_DF,"player_name",summarize,
  avg_age_in_career = mean(age,na.rm=TRUE),
  avg_player_height = mean(player_height,na.rm=TRUE),
  avg_player_weight = mean(player_weight,na.rm=TRUE),
  college   = unique(college),
  country = unique(country),
  draft_year = unique(draft_year),
  draft_round = unique(draft_round),
  draft_number = unique(draft_number),
  total_gp = sum(gp,na.rm=TRUE),
  avg_pts   = sum(season_total_pts,na.rm=TRUE)/sum(gp,na.rm=TRUE),
  avg_reb   = sum(season_total_reb,na.rm=TRUE)/sum(gp,na.rm=TRUE),
  avg_ast   = sum(season_total_ast,na.rm=TRUE)/sum(gp,na.rm=TRUE),
  avg_net_rating = mean(net_rating,na.rm=TRUE),
  avg_oreb_pct = mean(oreb_pct,na.rm=TRUE),
  avg_dreb_pct = mean(dreb_pct,na.rm=TRUE),
  avg_usg_pct = mean(usg_pct,na.rm=TRUE),
  avg_ts_pct = mean(ts_pct,na.rm=TRUE),
  avg_ast_pct = mean(ast_pct,na.rm=TRUE),
  modern_draft_round = unique(modern_draft_round))

write.csv(all_career_stats_DF,
          "Created_Data/all_career_stats.csv",row.names = FALSE)
```

A new classification tree is then created to predict the draft round in which player was selected.

```
career_data_column_list <- c("modern_draft_round","avg_age_in_career",
                        "avg_usg_pct","avg_pts","avg_reb","avg_ast","total_gp")
career_data_classification_tree_model <- rpart(modern_draft_round ~ .,
              method = "class", data = all_career_stats_DF[career_data_column_list])
rpart.plot(career_data_classification_tree_model,
           main = "Predicting Draft Round Number Using Career Data")
```



Transforming the NBA players' seasonal data into NBA players' career averages helps to balance the distribution of modern_draft_round more evenly: 41% for first-round players, 27% for second-round players, and 32% for undrafted players. This adjustment addresses the issue where first-round players disproportionately contained more data, due to their longer careers. Moreover, this updated model yields a more even prediction distribution: second-round outcomes occur 14% of the time, and undrafted outcomes occur 31% of the time.

```
career_data_classification_tree_model_predictions <-
  predict(career_data_classification_tree_model,
              all_career_stats_DF[career_data_column_list],type = "class")

career_data_confusion_matrix <-
  table(Prediction = career_data_classification_tree_model_predictions,
  Truth = all_career_stats_DF$modern_draft_round)

row_labels <- rownames(career_data_confusion_matrix)
updated_row_labels <- paste(row_labels,"(Predictions)",sep = " ")
column_labels <- colnames(career_data_confusion_matrix)
updated_column_labels <- paste(column_labels,"(Truth Values)",sep = " ")
rownames(career_data_confusion_matrix) <- updated_row_labels
colnames(career_data_confusion_matrix) <- updated_column_labels

knitr::kable(career_data_confusion_matrix,caption =
               "Confusion Matrix of Career Data Classification Tree Model")
```

Table 6: Confusion Matrix of Career Data Classification Tree Model

	Round 1 (Truth Values)	Round 2 (Truth Values)	Undrafted (Truth Values)
Round 1 (Predictions)	924	304	233
Round 2 (Predictions)	99	183	95
Undrafted (Predictions)	79	225	541

```
cat_labels <- c("Round 1","Round 2","Undrafted")
career_data_accuracies_DF <-
  CalculateAccuracies(career_data_confusion_matrix,cat_labels)

knitr::kable(career_data_accuracies_DF,caption =
               "Career Data Classification Tree Model Accuracies")
```

Table 7: Career Data Classification Tree Model Accuracies

	Accuracy
Round 1	0.6324435
Round 2	0.4854111
Undrafted	0.6402367
Overall	0.6142378

In this case, the confusion matrix and accuracy table clearly show the improved performance of this model. Although the accuracy for predicting first-round player data slightly decreased from roughly 63.27% to 63.24%, other categories saw an increase. The accuracy for second-round players increased from 45.32% to 48.54%, undrafted player accuracy surged from 46.11% to 64.02% and the overall accuracy climbed from 61.09% to 61.42%.

Let's take Example 1 as follows:

```
career_data_example_1_DF <- all_career_stats_DF[34,]
career_data_example_1_prediction <- predict(career_data_classification_tree_model,
                career_data_example_1_DF[career_data_column_list],type = "class")
cat("Example 1:","\n")
```

```
## Example 1:
```

```
cat("Input data:","\n")
```

```
## Input data:
```

```
head(career_data_example_1_DF[,!names(career_data_example_1_DF) %in% "modern_draft_round"])
```

```
##    player_name avg_age_in_career avg_player_height avg_player_weight college
## 34  AJ Griffin                19            198.12          99.79024    Duke
##    country draft_year draft_round draft_number total_gp avg_pts avg_reb avg_ast
## 34     USA       2022           1           16       72     8.9     2.1       1
##    avg_net_rating avg_oreb_pct avg_dreb_pct avg_usg_pct avg_ts_pct avg_ast_pct
## 34            1.5        0.026         0.08       0.174      0.577        0.07
```

Some key information about this player includes his 72 total games played, his average age throughout his career at 19, and his career average of 8.9 points scored per game.

His path down the tree can be detailed as follows:

1. He fails the first decision node, requiring a minimum of 136 total games played, and takes the right path.
2. He passes the second decision node, requiring a minimum of 22 total games played, and takes the left path.
3. He passes the third decision node, requiring a maximum average age through career of less than 22, and takes the left path.
4. He passes the fourth decision node, requiring a minimum career average points per game of 7.5, and takes the left path.
5. He arrives at the leaf node that 1% of the observations get to, and his prediction is given as: Round 1.

```
cat("Prediction:","\n")
```

```
## Prediction:
```

```
print(summary(career_data_example_1_prediction))
```

```
##   Round 1   Round 2 Undrafted
##         1         0         0
```

```
cat("Actual:","\n")
```

```
## Actual:
```

```
cat(career_data_example_1_DF$modern_draft_round,"\n")
```

```
## Round 1
```

Let's now take a look at Example 2:

```
cat("\n")
```

```
career_data_example_2_DF <- all_career_stats_DF[15,]
career_data_example_2_prediction <- predict(career_data_classification_tree_model,
                career_data_example_2_DF[career_data_column_list],type = "class")
cat("Example 2:","\n")
```

```
## Example 2:
```

```
cat("Input data:","\n")
```

```
## Input data:
```

```
head(career_data_example_2_DF[,!names(career_data_example_2_DF) %in% "modern_draft_round"])
```

```
##      player_name avg_age_in_career avg_player_height avg_player_weight  college
## 15 Aaron Wiggins              23.5            194.31          86.18248 Maryland
##    country draft_year draft_round draft_number total_gp avg_pts avg_reb avg_ast
## 15     USA       2021           2           55      120   7.425    3.25   1.225
##    avg_net_rating avg_oreb_pct avg_dreb_pct avg_usg_pct avg_ts_pct avg_ast_pct
## 15           -3.3        0.045       0.1035       0.146     0.5815      0.0835
```

Some key information about this player includes his 120 total games played, and his average age throughout his career at 23.5.

His path down the tree can be detailed as follows:

1. He fails the first decision node, requiring a minimum of 136 total games played, and takes the right path.
2. He passes the second decision node, requiring a minimum of 22 total games played, and takes the left path.
3. He fails the third decision node, requiring a maximum average age through career of less than 22, and takes the right path.

4. He fails the fourth decision node, requiring a minimum average age through career of 32, and takes the right path.
5. He passes the fifth decision node, requiring a maximum average age through career of less than 24, and takes the left path.
6. He arrives at the leaf node that 9% of the observations get to, and his prediction is given as: Round 2.

```
cat("Prediction:","\n")
```

```
## Prediction:
```

```
print(summary(career_data_example_2_prediction))
```

```
##   Round 1   Round 2 Undrafted
##         0         1         0
```

```
cat("Actual:","\n")
```

```
## Actual:
```

```
cat(career_data_example_2_DF$modern_draft_round,"\n")
```

```
## Round 2
```

Predicting NBA Player Usage Rate Using Best Subset Selection and PCA

The modified seasonal dataset (all_seasons_modified_DF) is used again for this task. The usage rate (usg_pct) is selected as the response variable, while the other five variables used in the prior questions are selected as explanatory variables. Initially, the input and output data are extracted. Then, the categorical variable modern_draft_round undergoes one-hot encoding, separating into three variables: round_1, round_2, and undrafted. Each variable holds a value of 1 if the player was drafted in that specific round, and 0 otherwise.

The squared terms are added to the input dataset for age, usg_pct, pts, reb, and ast. The data is then fed into a best subset model to find the combination of variables that best explain the variation of the response variable.

```
library(leaps)

six_most_important_variables_list <-
  c("modern_draft_round","age","usg_pct","pts","reb","ast")
output_variable <- c("usg_pct")
input_variables_list <- setdiff(six_most_important_variables_list,output_variable)

model_input_data_DF <- all_seasons_modified_DF[,input_variables_list]

model_input_data_DF <- mutate(model_input_data_DF,
      round_1 = ifelse(modern_draft_round == "Round 1",1,0),
      round_2 = ifelse(modern_draft_round == "Round 2",1,0),
      undrafted = ifelse(modern_draft_round == "Undrafted",1,0))
model_input_data_DF <-
  model_input_data_DF[,!names(model_input_data_DF) %in% "modern_draft_round"]
model_input_data_DF$age_squared <- model_input_data_DF$age^2
model_input_data_DF$pts_squared <- model_input_data_DF$pts^2
model_input_data_DF$reb_squared <- model_input_data_DF$reb^2
model_input_data_DF$ast_squared <- model_input_data_DF$ast^2

model_data_DF <- model_input_data_DF
model_data_DF$usg_pct <- all_seasons_modified_DF$usg_pct

subset_regression_model <- regsubsets(usg_pct ~ .,data = model_data_DF,nbest = 1,
really.big = TRUE)
```

```
## Reordering variables and trying again:
```

```
subset_regression_model_summary <- summary(subset_regression_model)
best_regression_model_index <- which.max(subset_regression_model_summary$adjr2)
best_regression_model_coefficients <-
  coef(subset_regression_model,best_regression_model_index)
best_regression_model_coefficients <-
  best_regression_model_coefficients[best_regression_model_coefficients != 0]
best_regression_model_variables <- names(best_regression_model_coefficients)[-1]
best_regression_model_adjusted_r_squared_value <-
subset_regression_model_summary$adjr2[best_regression_model_index]

cat("\n")
```

```
cat("Best model variables:","\n")
```

```
## Best model variables:
```

```
cat(best_regression_model_variables,"\n")
```

```
## age pts reb ast round_1 pts_squared reb_squared ast_squared undrafted
```

The best model's variables: age, pts, reb, ast, round_1, pts_squared, reb_squared, ast_squared, and undrafted are then fed into a linear regression model.

```
best_regression_model_variables_and_usg_pct <-
  c("usg_pct",best_regression_model_variables)

best_regression_model_variables_and_usg_pct_DF <-
  model_data_DF[,best_regression_model_variables_and_usg_pct]

best_subset_lm_model <- lm(usg_pct ~ .,
  data = best_regression_model_variables_and_usg_pct_DF)

best_subset_lm_model_SUMMARY <- summary(best_subset_lm_model)

cat("Best Subset Model Summary")
```

```
## Best Subset Model Summary
```

```
print(best_subset_lm_model_SUMMARY)
```

```
##
## Call:
## lm(formula = usg_pct ~ ., data = best_regression_model_variables_and_usg_pct_DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.17577 -0.02194 -0.00231  0.01884  0.83093
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.958e-01  2.291e-03  85.467  < 2e-16 ***
## age         -1.117e-03  7.726e-05 -14.453  < 2e-16 ***
## pts          9.962e-03  2.506e-04  39.757  < 2e-16 ***
## reb         -1.847e-02  5.030e-04 -36.728  < 2e-16 ***
## ast         -1.153e-02  6.418e-04 -17.966  < 2e-16 ***
## round_1      4.495e-03  8.590e-04   5.233 1.69e-07 ***
## pts_squared -3.999e-05  8.282e-06  -4.829 1.39e-06 ***
## reb_squared  1.029e-03  4.042e-05  25.468  < 2e-16 ***
## ast_squared  9.598e-04  7.242e-05  13.254  < 2e-16 ***
## undrafted   -9.890e-04  1.039e-03  -0.952    0.341
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03745 on 12834 degrees of freedom
## Multiple R-squared:  0.5112, Adjusted R-squared:  0.5108
## F-statistic:  1491 on 9 and 12834 DF,  p-value: < 2.2e-16
```

From the model summary, it can be seen that the residual values (the difference between true and predicted values) vary from -0.17577 to 0.83093, which suggests variability in the model's accuracy for different observations. However, the median difference is -0.00231. The coefficients for the linear regression model are shown in the Estimate column. Positive values signify a positive relationship with usg_pct, whereas negative values indicate a negative relationship. The p values within the Pr(>|t|) column indicate that all coefficients

for the variables are statistically significant, except for the undrafted coefficient which exceeds the commonly used threshold of 0.05. The p-value associated with the F-statistic of 1491 indicates that the model's variables collectively describe the variance of modern_draft_round beyond random variance. To be more precise, the adjusted R-squared value of 0.5108 implies that roughly 51.08% of the variance of usg_pct is explained.

Now let's examine the predicted outcomes of two examples:

```
example_1_DF <- model_data_DF[1,]
example_1_prediction <-
  predict(best_subset_lm_model,example_1_DF[,!names(example_1_DF) %in% "usg_pct"])
cat("Example 1:","\n")
```

```
## Example 1:
```

```
cat("Input data:","\n")
```

```
## Input data:
```

```
head(example_1_DF[,!names(example_1_DF) %in% "usg_pct"])
```

```
##   age pts reb ast round_1 round_2 undrafted age_squared pts_squared reb_squared
## 1  22 3.9 1.5 2.4       0       1         0         484       15.21        2.25
##   ast_squared
## 1        5.76
```

```
cat("Predicted usg_pct:",example_1_prediction,"\n")
```

```
## Predicted usg_pct: 0.1619788
```

```
cat("Actual usg_pct:",example_1_DF$usg_pct,"\n")
```

```
## Actual usg_pct: 0.169
```

```
cat("\n")
```

```
example_2_DF <- model_data_DF[500,]
example_2_prediction <-
  predict(best_subset_lm_model,example_2_DF[,!names(example_1_DF) %in% "usg_pct"])
cat("Example 2:","\n")
```

```
## Example 2:
```

```
cat("Input data:","\n")
```

```
## Input data:
```

```
head(example_2_DF[,!names(example_1_DF) %in% "usg_pct"])
```

```
##     age  pts reb ast round_1 round_2 undrafted age_squared pts_squared
## 500  35 16.1 8.4   1       1       0         0        1225      259.21
##     reb_squared ast_squared
## 500       70.56           1
```

```
cat("Predicted usg_pct:",example_2_prediction,"\n")
```

```
## Predicted usg_pct: 0.2181659
```

```
cat("Actual usg_pct:",example_2_DF$usg_pct,"\n")
```

```
## Actual usg_pct: 0.244
```

At seen above, the combination of the linear model with the best subset appears to somewhat accurately predict the usage rate.

Next, the input data is centered and standardized before applying principal component analysis. The resulting data is then analyzed and visualized to see whether a linear model with fewer dimensions could accurately predict usage rates.

```
model_input_data_PCA <- prcomp(model_input_data_DF,center = TRUE,scale = TRUE)
model_input_data_PCA_weights <- model_input_data_PCA$rotation
model_input_data_PCA_center <- model_input_data_PCA$center
model_input_data_PCA_scale <- model_input_data_PCA$scale

model_input_data_PCA_SUMMARY <- summary(model_input_data_PCA)

print(model_input_data_PCA_SUMMARY)
```

```
## Importance of components:
##                            PC1    PC2    PC3    PC4    PC5     PC6     PC7
## Standard deviation     2.0230 1.4185 1.2783 1.1712 1.0964 0.72232 0.29557
## Proportion of Variance 0.3721 0.1829 0.1486 0.1247 0.1093 0.04743 0.00794
## Cumulative Proportion  0.3721 0.5550 0.7035 0.8283 0.9375 0.98497 0.99291
##                            PC8     PC9    PC10      PC11
## Standard deviation     0.22996 0.14522 0.06341 4.732e-15
## Proportion of Variance 0.00481 0.00192 0.00037 0.000e+00
## Cumulative Proportion  0.99772 0.99963 1.00000 1.000e+00
```

As expected, the initial principal components account for the majority of the variance present in the dataset, with each subsequent component making a diminishing contribution. The choice of how many principal components to retain can vary depending on the application, often determined by a selected cumulative variance threshold. In this case, a threshold of 0.95 is chosen as model complexity is not an issue.

```
variance_threshold <- 0.95
cumulative_variance_list <- model_input_data_PCA_SUMMARY$importance["Cumulative Proportion",]
PCA_num_components <- which(cumulative_variance_list >= variance_threshold)[1]
PCA_variables_used <- colnames(model_input_data_PCA$x[,1:PCA_num_components])

cat("PCA variables used (threshold = ",variance_threshold,"):",sep="")
```

```
## PCA variables used (threshold = 0.95):
```

```
cat("\n")
```

```
cat(PCA_variables_used,"\n")
```

```
## PC1 PC2 PC3 PC4 PC5 PC6
```

In this case, selecting the first six principal components meets the threshold of explaining at least 95% of the variance captured in the dataset.

```
model_PCA_input_data_DF <-
  as.data.frame(model_input_data_PCA$x[,1:PCA_num_components])
model_PCA_data_DF <- model_PCA_input_data_DF
model_PCA_data_DF$usg_pct <- all_seasons_modified_DF$usg_pct

model_PCA_data_lm_model <- lm(usg_pct ~ ., data = model_PCA_data_DF)
model_PCA_data_model_summary <- summary(model_PCA_data_lm_model)
cat("PCA Model Summary")
```

```
## PCA Model Summary
```

```
print(model_PCA_data_model_summary)
```

```
##
## Call:
## lm(formula = usg_pct ~ ., data = model_PCA_data_DF)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.16162 -0.02383 -0.00299  0.01985  0.84595
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.1846408  0.0003473 531.717   <2e-16 ***
## PC1          0.0140294  0.0001717  81.730   <2e-16 ***
## PC2         -0.0060738  0.0002448 -24.809   <2e-16 ***
## PC3         -0.0068445  0.0002717 -25.196   <2e-16 ***
## PC4          0.0003293  0.0002965   1.111    0.267
## PC5          0.0003670  0.0003167   1.159    0.247
## PC6          0.0263530  0.0004808  54.815   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.03935 on 12837 degrees of freedom
## Multiple R-squared:   0.46,  Adjusted R-squared:  0.4598
## F-statistic:  1823 on 6 and 12837 DF,  p-value: < 2.2e-16
```

In this case, the residual values vary from -0.16162 to 0.84595, once again implying variability in the model's accuracy for various observations. The median residual is -0.00299. The linear regression model coefficients are shown in the Estimate column. The p values within the Pr(>|t|) column indicate that all coefficients for the first four, as well as the sixth principal components are statistically significant. The p-value associated with the F-statistic of 1823 indicates that the model's variables as a whole describe the variance of modern_draft_round beyond random variance. In other words, the adjusted R-squared value of 0.4598 implies that roughly 45.98% of the variance of usg_pct is explained.

Since the magnitude of the median residual value is higher, and the adjusted R-squared value is lower for the PCA and linear model combination, this suggests that the best subsets and linear regression model combination is better.

Now let's examine the predicted outcomes of two examples:

```
example_1_DF <- model_data_DF[1,]
example_1_scaled_DF <- scale(example_1_DF[,!names(example_1_DF) %in% "usg_pct"],
            center = model_input_data_PCA_center,scale = model_input_data_PCA_scale)
example_1_PCA_DF <-
  as.data.frame(t(model_input_data_PCA_weights%*%t(example_1_scaled_DF)))
colnames(example_1_PCA_DF) <- PCA_variables_used
example_1_prediction <-
  predict(model_PCA_data_lm_model,example_1_PCA_DF[,1:PCA_num_components])
cat("Example 1:","\n")
```

```
## Example 1:
```

```
cat("Input data:","\n")
```

```
## Input data:
```

```
head(example_1_DF[,!names(example_1_DF) %in% "usg_pct"])
```

```
##   age pts reb ast round_1 round_2 undrafted age_squared pts_squared reb_squared
## 1  22 3.9 1.5 2.4       0       1         0         484       15.21        2.25
```

```
##   ast_squared
## 1        5.76
```

```
cat("Predicted usg_pct:",example_1_prediction,"\n")
```

```
## Predicted usg_pct: 0.2057146
```

```
cat("Actual usg_pct:",example_1_DF$usg_pct,"\n")
```

```
## Actual usg_pct: 0.169
```

```
cat("\n")
```

```
example_2_DF <- model_data_DF[500,]
example_2_scaled_DF <- scale(example_2_DF[,!names(example_2_DF) %in% "usg_pct"],
            center = model_input_data_PCA_center,scale = model_input_data_PCA_scale)
example_2_PCA_DF <-
    as.data.frame(t(model_input_data_PCA_weights%*%t(example_2_scaled_DF)))
colnames(example_2_PCA_DF) <- PCA_variables_used
example_2_prediction <-
  predict(model_PCA_data_lm_model,example_2_PCA_DF[,1:PCA_num_components])
cat("Example 2:","\n")
```

```
## Example 2:
```

```
cat("Input data:","\n")
```

```
## Input data:
```

```
head(example_2_DF[,!names(example_2_DF) %in% "usg_pct"])
```

```
##     age  pts reb ast round_1 round_2 undrafted age_squared pts_squared
## 500  35 16.1 8.4   1       1       0         0        1225      259.21
##     reb_squared ast_squared
## 500       70.56           1
```

```
cat("Predicted usg_pct:",example_2_prediction,"\n")
```

```
## Predicted usg_pct: 0.1644906
```

```
cat("Actual usg_pct:",example_2_DF$usg_pct,"\n")
```

```
## Actual usg_pct: 0.244
```

At seen above, the combination of the linear model with the principal component analysis techniques appear to be significantly less accurate at predicting the usage rate than the linear model with best subset regression.

Steps to be Taken to Ensure Analysis Reproducibility

Several steps would be taken to ensure that the analysis would be reproducible and easy to evaluate by others. Firstly, extensive automation of the code would be prioritized. Key aspects of the analysis would be encapsulated into user-friendly functions, such as dataset creation, generating ggpairs plots, and constructing classification trees. These functions would only consist of necessary inputs, like relative file paths for data access, loaded datasets, input/output variables, and important model parameters.

Additionally, rigorous testing would be conducted to discover potential errors that could cause the program to crash. Comprehensive error-handling mechanisms would be integrated to quickly deal with any issues that arise during runtime. All programming scripts and data versions would be carefully documented and shared on GitHub, accompanied by descriptive commit messages so that users could easily follow the evolution of the scripts.

The next step would be to make the code as interpretable as possible by utilizing informative variable names and adding detailed comments explaining key parts of the program. A README file would be created containing several sections. This would cover project objectives, installation guidelines for required packages and dependencies, specifics of R language versions used, and a comprehensive list of necessary packages with version details. It would also explain the code, data transformations and step-by-step instructions for running the analysis using the defined functions. Finally, it would outline guidelines for project enhancement and reporting script issues.

The packages would be managed using tools like the renv package, which allows an R project's environment to be seamlessly saved and reproduced across other machines. Furthermore, an annual comprehensive report would be generated and shared on GitHub to identify emerging trends in basketball analytics. Future project expansion plans may involve automating analysis and report generation using task automation tools like Windows Task Manager.

Ethical Concerns From the use of Dataset

Working with datasets often triggers ethical concerns due to the potential exposure of personal information. Residual disclosure in data analytics occurs when sensitive details about an individual inadvertently surface, even when precautions are taken to protect the data. This typically happens when external information is linked back to an individual from the dataset. While this dataset primarily consists of player performance statistics, as it is analyzed to uncover relationships between variables and develop predictive models, biased expectations regarding a player's career trajectory could emerge. For instance, a talented but shorter player might be overlooked due to the statistical advantages generally attributed to taller players, such as scoring and rebounding. The same bias is often seen in undrafted players in comparison to lottery picks or first-round players.

Take the case of Fred VanVleet, an NBA player initially passed over by several teams due to his relatively short stature of 6 feet. Despite being undrafted, VanVleet excelled with the Toronto Raptors, setting franchise records, becoming an all-star and NBA champion, and signing the most expensive deal for an undrafted player in NBA history.

This being said, external data sources could still easily be intertwined with performance data, potentially exposing private information players prefer to keep confidential. For example, a decline in performance might spark unwarranted speculation about a player's health or personal issues, especially when such information is readily accessible on social media. Consent issues could also arise when collecting data on players as they may not be aware of the use of their statistics (i.e. physical or performance traits) within data analysis or predictive modelling.

Also, the data from certain players may cause the general public or social media outlets to misjudge players, as many casual fans pay attention to the measurable stats, without considering a player's impact beyond the stat sheet (i.e. energy, effort, leadership). This could cause certain players to feel self-conscious and could even cause disparities in salaries.

Finally, there could be inaccuracies in player data, which could skew perceptions of their potential or style of play. This could even impact the opposing team's strategies and lineup, or affect teams' integrities. A good example is Kevin Durant, who has been listed as 6'9 or 6'10 throughout his career until several people realized that he appeared to be closer to 7 feet.