

# Applying Machine Learning- Supervised Learning Techniques for Tennis Players Dataset Analysis

Moaiad Ahmad Khder, Samah Wael Fujo

Department of Computer Science – Applied Science University-Bahrain

e-mail: moaiad.khder@asu.edu.bh

Data Scientist – Nasser Artificial Intelligence Research and Development Nasser

Centre - Nasser Vocational Training Center-Bahrain

e-mail: samah.fujo@nvtc.edu.bh

## Abstract

*ATP Tennis stands for the “The Association of Tennis Professionals” which is the primary governing body for male tennis players. ATP was formed in Sep 1972 for professional tennis players. A study has been done on tennis players’ datasets to implement supervised machine learning techniques to illustrate match data and make predictions. An appropriate dataset has been chosen, data cleaning has been implemented to extract anomalies, data is visualized via plotting methods in R language and supervised machine learning models applied. The main models applied are linear regression and decision tree. Results and predictions have been extracted from the applied models. In the linear regression model, the correlation is calculated to find the relation between dependent and independent variables, furthermore the results and prediction are extracted from the linear regression model. Also, three hypotheses are applied for multiple linear regression model. The decision tree modeled the best of 3 or best of 5 sets of matches and predicted which set of matches would be considered best.*

**Keywords:** Machine Learning, supervised learning, linear regression, decision tree, R language, Tennis, ATP.

## 1 Introduction

Tennis is a sport which stems from the racquet sports category. This sport can be played against one player or between double players. The tennis players face each other on a rectangular tennis court field separated with a net in the middle across its width. Tennis can be played on a multitude of surfaces such as hard, clay, grass or carpet, which are used in many tournaments. If the players made two attempts, which in tennis terminology is called a server legal, the players go back and forth, hitting the ball before a player scores a point to win the rally (Sipko, 2015).

This research aims to implementing supervised machine learning models on a tennis match dataset (tennis players and tennis matches data gather throughout the years) to analyze, extract relevant knowledge and obtain predictions. The main objectives are:

- To obtain and analyze a tennis players dataset.
- To apply supervised machine learning techniques: linear regression and decision tree.
- To extract relevant knowledge from analyzing the dataset.

## **2 Machine Learning Models Background**

Supervised Machine Learning (SML) is the quest for algorithms that reason from externally supplied instances to generate broad hypotheses, which subsequently make predictions about future instances.(Burkart & Huber, 2021; Singh et al., 2016; T Akinsola et al., 2017)

### **2.1 Linear regression**

The correlation coefficient will be used to calculate the relationship between the two variables. It is given the strength and direction of the relationship between the two variables. It remains also important to note that the correlation coefficient between the two variables measure only the relationship. Regression analysis is a mathematical method or technique that examines the relationship between the objective (dependent) and the predictor (independent variable). In the study of regression analysis, variables can be categorized into two types, dependent variable and independent variables (Damanik et al., 2019) (Pant & R. S. Rajput, 2019). The values of the dependent variable resulted from changes in the values of the dependent variables. Depending on the number of variables available, the regression analysis is split into two groups: simple linear regression and multiple linear regression. A simple linear regression model describes the linear relationship between two variables; furthermore, a multiple linear regression model describes the linear relationship between a single dependent variable and several independent variables (Pant & R. S. Rajput, 2019) .

### **2.2 Decision tree**

The decision tree is a flexible algorithm for machine learning, capable of both regression and classification functions. They are incredibly efficient algorithms that suit complex datasets. Furthermore, a decision tree is a fundamental aspect of "random forests," one of the most potent algorithms for machine learning nowadays (Johnson, 2022).

Decision-tree is the most effective and fastest data mining technology usually used in prediction and data analysis (Es-sabery & Hair, 2019). A decision-tree method turns a very significant fact into a decision tree representing the rules. For natural language, the rules can be easily interpreted. The essential advantage of using a decision tree is its capability to break down complicated decision-

making processes to be easier so that decision-making can better define problem solutions by translating data type [tables] into a decision-tree model, converting the decision-tree model into a rule. Decision-trees are also useful when exploring data, namely attempting to find hidden patterns and relationships with a target variable between several potential input variables (Damanik et al., 2019).

### 3 Methodology

There are several key phases to any project, the phases start by first collecting the appropriate data set, clean the data, visualizing the data, building the machine learning models, evaluation and analysis of the results obtained from the mode. Fig 1 shows the methodology phases.



Fig 1 Methodology Phases

#### 3.1 Data collection

A dataset is a set of data grouped together for analysis purposes. It contains several features that describe the information gathered. The dataset collected for this project sourced from (Sackmann, 2020). The obtained dataset contains information about tennis players since 1968 until 2020. Some of the features include players names, rankings, tourney information, age and other details discussed later in the design and implementation phases. The Fig 2 samples the data used in the dataset from the website (ATP Tour, 2022).

Ranking ^	Move ^	Country ^	Player ^	Age ^	Points ^	Tourn Played ^	Points Dropping ^	Next Best ^
1	-		Novak Djokovic	32	10,220	18	45	0
2	-		Rafael Nadal	33	9,850	18	360	0
3	-		Dominic Thiem	26	7,045	21	1,000	90
4	-		Roger Federer	38	6,630	16	600	0
5	-		Daniil Medvedev	24	5,890	23	45	45

Fig 2 Dataset preview

##### 3.1.1 Data dictionary

In the ATP Tennis Player dataset, after data cleaning, it contains forty-two columns. We have chosen some columns that need to be understood what they mean to help with the analysis process (“Historical Dictionary of Tennis,” 2012; Jefferys, 2012; Lake, 2012). The explanation shows in the Table 1.

Table 1. Data set features explanation

Feature	Details
tourney_id	"a unique identifier for each tournament"
tourney_name	"The name of a tourney"
Surface	"Tennis is played on a variety of surfaces and each surface has its own characteristics which affect the playing style of the game. There are four main types of courts depending on the materials used for the court surface: clay courts, hard courts, grass courts and carpet courts. "
Draw size	"number of players in the draw, often rounded up to the nearest power of 2"
Tourney level	"Name of ATP tennis series which are "ATP Tour 250/500 Series", "Masters", "Grand Slam", and "Tour Final". "
Tourney date	"Date of tourney, usually the Monday of the tournament week. "
Winner id	"the player_id used for the winner of the match"
Winner name	The name of the winner
Winner hand	"Winner hand used in the game: left or right "
Winner height	"Winner's height in centimeters, where available"
Winner ioc	"Winner's three-character country code"
Winner age	"age, in years, as of the tourney_date"
Loser id	"the player_id used for the loser of the match"
Loser name	"Winner hand used in the game: left or right"
Loser hand	"loser hand used in the game: left or right"
Loser height	"Loser's height in centimeters, where available"
Loser ioc	"loser's three-character country code"
Loser age	"age, in years, as of the tourney_date"
Score	"Score points, method of tracking progress of a match. A match consists of points, game and sets"
Best of	"3' or '5', indicating the number of sets for this match Set: A unit of scoring. A set consists of games and the first player to win six games with a two-game advantage wins the set. In most tournaments a tiebreak is used at six games all to decide the outcome of a set. "
Round	"Round of matches from first round to the final which are "First Round, with 128 players (sixty-four matches) → R128" "Second Round, with 64 players (thirty-two matches) → R64" "Third Round, with 32 players (sixteen matches) → R32" "Fourth Round, with 16 players (eight matches) → R16" "Quarterfinals, with 8 players (four matches) → QF" "Semifinals, with 4 players (two matches) → SF" "Final, with the last two players playing for the title → F" "
Minutes	"match length, where available"
w_ace	"Winner's number of aces. A serve that the returner does not even touch with her racquet. An ace wins the point immediately for the server"
w_df	"winner's number of doubles faults. Two serving faults in a row in one point, causing the player serving to lose the point
w_svpt	Winner's number of serve points. Serve: the starting stroke of each point. The ball must be hit into the opponent's service box, specifically the box's half that is diagonally opposite the server"
w_1stIn	"Winner's number of first serves made. A first serve is made when there has been no fault on the point"
w_1stWon	"winner's number of first-serve points won"
w_2ndWon	"winner's number of second-serve points won"
w_SvGms	"Second serve occurs when there has already been one fault on the point. "
w_bpSaved	"winner's number of serve games" "winner's number of break points saved" "a point which allows the receiving player to break the service of the server"

w_bpFaced	"winner's number of break points faced"
l_ace	"loser's number of aces."
l_df	"loser's number of doubles faults"
l_svpt	"loser's number of serve points"
l_1stIn	"loser's number of first serves made"
l_1stWon	"loser's number of first-serve points won"
l_2ndWon	"loser's number of second-serve points won"
l_SvGms	"loser's number of serve games"
l_bpSaved	"loser's number of break points saved"
l_bpFaced	"loser's number of break points faced"
Winner rank	"loser's ATP or WTA rank, as of the tourney_date, or the most recent ranking date before the tourney_date."
	"Or A hierarchical listing of players based on their recent achievements. Used to determine qualification for entry and seeding in tournaments."
Loser rank	"loser's ATP or WTA rank, as of the tourney_date, or the most recent ranking date before the tourney_date"

### 3.2 Data cleaning

Data cleaning is one of the most crucial steps to be taken into consideration before implementing any model on large sets of data. This step can be challenging since the enormous size of the data and its rapid growth. Data cleaning is the process of eliminating and correcting false data, empty values, or corrupt data to remove impurities (Fujo et al., 2022). Corrupt data can lead to inaccurate modeling and prediction of data which will lead to false results. Filtering the data is also included in the process of data cleansing. Cleaning filters out data outliers including duplicate, null, poorly formatted, and incorrect records. The process of data cleaning implemented on our dataset is explained in the implementation phase.

### 3.4 Data visualization

Data visualization defines as the implementation of modern visualization methods to model and illustrate data and relationships within. Visualization methods include applying techniques that project real time changes. Data visualization is important to understand patterns existing within the dataset and further understand the data before implementing machine learning models for predictions. Visuals tend to explain data better than numeric interpretations. Some visualization libraries used during our implementation will be discussed in detail below.

### 3.5 Build the ML model

For this case study, two supervised machine learning models will be implemented which are multilinear regression and decision tree. Each model focuses on different aspects of the dataset, represents and models. The next phase discusses more on how the models will implemented, libraries and results.

### 3.6 Evaluation and Analysis

The final phase of methodology is model evaluation and analysis which highlights the main observations resulting from the implemented models. Observations such as plotting, visualizing the graphing the predictions.

## 4 Data preprocessing and visualization

### 4.1 Import data

After the necessary packages have been loaded, now it is time to import the data set, there are 65 files of ATP tennis matches that have been collected, but our focusing will only be on the ATP matches from 1986 to 2020. Thus the number of files that have been imported to the workspace is 52 CSV files.

### 4.2 Data cleaning

In this section, the data will be clean and normalized to be ready for analyzing it and extract good knowledge from it. The data cleaning has been done with several steps. Firstly, as the data is separated due to different years in different files, it will be hard to manage them. So the first step is to combine all CSV files in one data frame by using *rbind.fill()* function. After combining the data, we got 183471 rows; for sure, not all of the details are necessary, so it should be analyzed to clean and apply the needed change on data.

Starting with the date when check the tourney date noticed that it is listed as normal numeric which it requires a data type change, so the first step is to convert the tourney date from integer to date and split them by dash using *transform()* function.

Now coming to the removing unnecessary data step, when understanding the data, notice that the tourney name "Davis Cup" is almost incomplete, and the "Davis Cup" play doesn't match the rest of the data so it will be removed from the data frame.

Now removing Challenger level matches where *tourney\_level* equal C. they should not be in the data set, possibly a mistake because it appears only in some years, this may cause a mistics in the result of the prediction. Since London Olympics is not an international so it will be removed from the data set , and Beijing Olympics will be removed too.

An extra character has been found in the feature 's-Hertogenbosch, for easier manipulation the column has to be renamed to remove the extra character.

Rows of incomplete data has to be removed prior to implementing the models because outliers could affect the final prediction and give inaccurate results.

There are columns that were found to be irrelevant to our case and will most probably not be used in any of the models. For the most accurate results, unused columns have to be removed such as "match\_num, winner\_entry, loser\_entry, winner\_rank\_points, loser\_rank\_points".

Additionally, the *winner\_seed* and *loser\_seed* in leiu features of world winner rank/loser rank have to be excluded.

Data of players retired have found to be irrelevant because the main object is to find the ranks of current tennis players. So, rows where it belongs to retired players have to be filtered out.

Walkovers or players who quit during the game service no purpose to our analysis because we only require winners/losers who completed the match. Filtering the forfeiters is mandatory.

Some feature names are ambiguous, for better understanding the columns are renamed such as A stands for ATP Tour, M stands for Masters, G stands for Grand Slam and F stands for Tour Final. Moreover, the winner\_ht and loser\_ht columns renamed to winner\_height and loser\_height.

The last step is redefining the variable, best of, draw size and tourney level variable form. That of these variables was interpreted as a type of character or integer, whether they really are factors.

Also defining the round levels, as the tournament rounds have a legacy order. The last stage of the round is RR, for clarify. This means "round robin" and displays matches played in a round robin tournament. The only matches which fulfill this requirement are the ATP World Tour Finals that are held every year.

After data cleaning, it's worth to view the structure of cleaned data by three ways: str and glimpse and view functions.

After Data Cleaning, the dataset is ready to be analyzed. Function *str* and *glimpse* used to view the structure of cleaned dataset. the *glimpse()* function used to make it possible to see every column in a data frame

### 4.3 Data visualization and analysis

The "ATP Tennis Matches" data set includes a data collection from 1986 to 2020 for the men's ATP circuit. This section provides some analysis and visualization for ATP tennis matches.

- **Countries and Players**

In this project we observe the USA is the outright winner with 168 players. Spain comes in second place, where it holds 87 players. The third place is France, with 86 players the other countries shown in the Fig 8. The Fig 3 shows only the top of thirty countries.

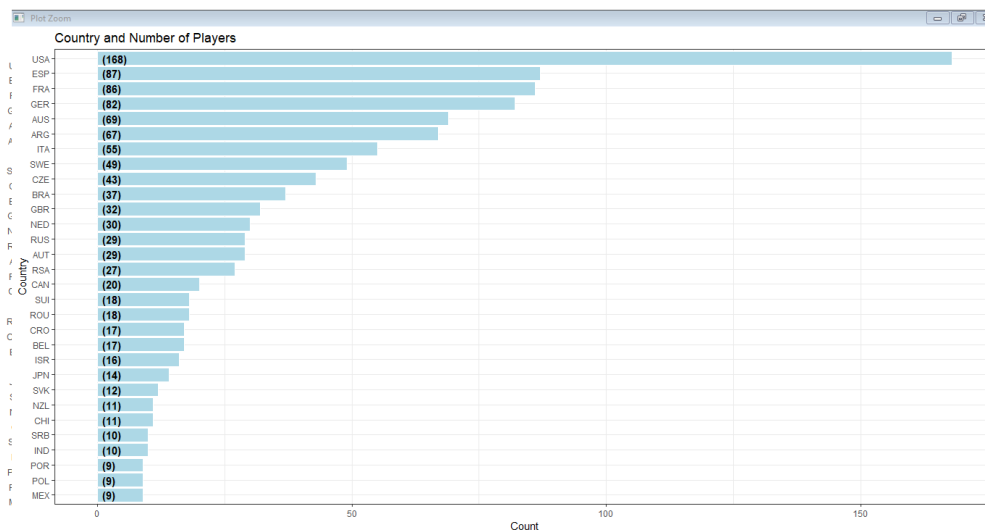


Fig 3. Countries and Players

- **Rankings and Year**

Display the players and the number one ranking they held each year. After that, create a bar chart that indicates the number of years they have at least the number one. The Fig 4 shows the top 10 winner players and count how many times got No 1 Ranking for each year. The players could be lower than number one in the same year, which shows the players' longevity and persistence. The usual names like Roger Federer, Rafael Nadal, Novak Djokovic, Peta Sampras, and Andre Agassi have dominated tennis for the past 52 years.

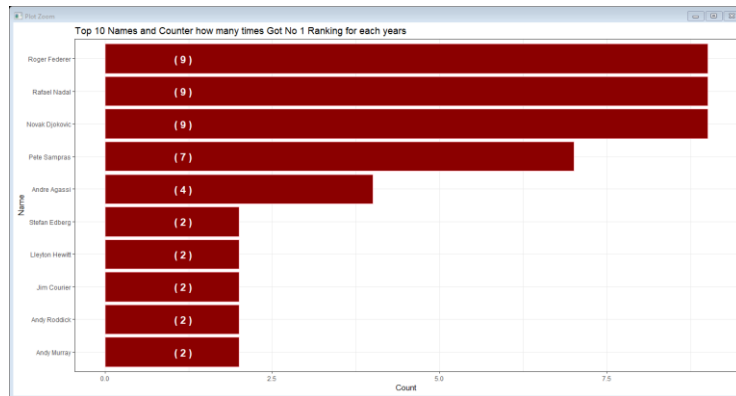


Fig 4. Top 10 players with ranking No 1

- **Number of First Ranking Tennis Players from 2005 to 2020:**

There is no complete domination from years 2005 - 2020 except for 2006, 2007, 2015, and 2019. For 2015 and 2019 dominated by Novak Djokovic.

The ranking data are shown in the Fig 5:

winner_name	year	winner_name	year
1 Novak Djokovic	2020	11 Novak Djokovic	2015
2 Rafael Nadal	2020	12 Novak Djokovic	2014
3 Novak Djokovic	2019	13 Rafael Nadal	2014
4 Novak Djokovic	2018	14 Novak Djokovic	2013
5 Rafael Nadal	2018	15 Rafael Nadal	2013
6 Roger Federer	2018	16 Novak Djokovic	2012
7 Andy Murray	2017	17 Roger Federer	2012
8 Rafael Nadal	2017	18 Novak Djokovic	2011
9 Andy Murray	2016	19 Rafael Nadal	2011
10 Novak Djokovic	2016	20 Rafael Nadal	2010

Fig 5.. Ranking data from 2005- 2020

- **Adelaide Winners**

The following Fig 6 shows the Adelaide winners:



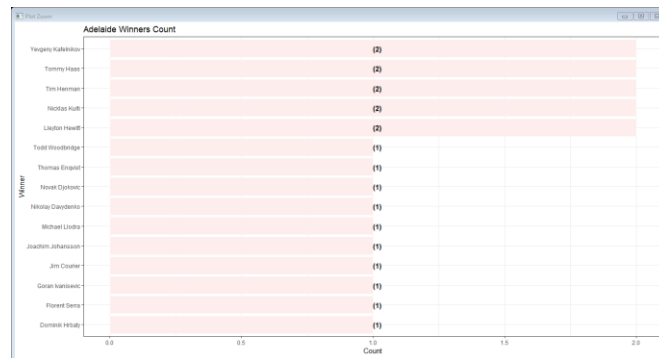


Fig 6. "Adelaide Winners"

- **Distribution of Age of Adelaide Winners:**

```
> summary(Adelaide$winner_age)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 17.00  20.00  23.00  22.95  25.25  28.00
```

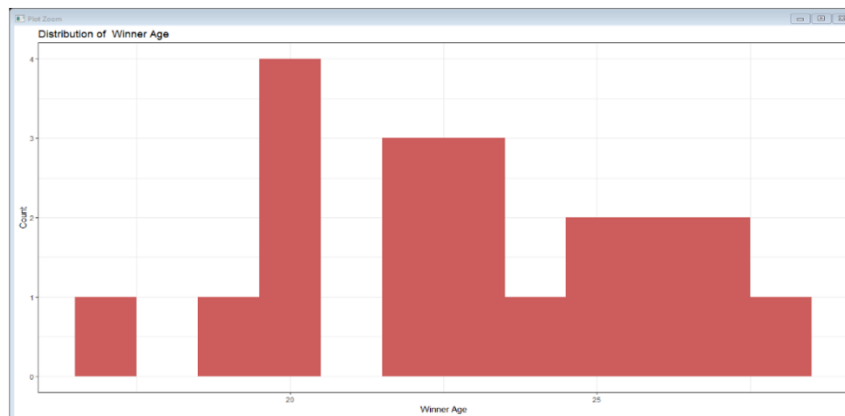


Fig 7. Age of Adelaide Winners

As you can see in the Fig 7, the minimum age of tourney Adelaide is 17, and the maximum period is 28, while the middle age is between 22 and 23, and the most repeated age is 20. The same process can be applied to the other tourneys to see the difference in ages.

- **Surface**

Fig 8 shows the number of matches on different surfaces are shown in a bar plot.

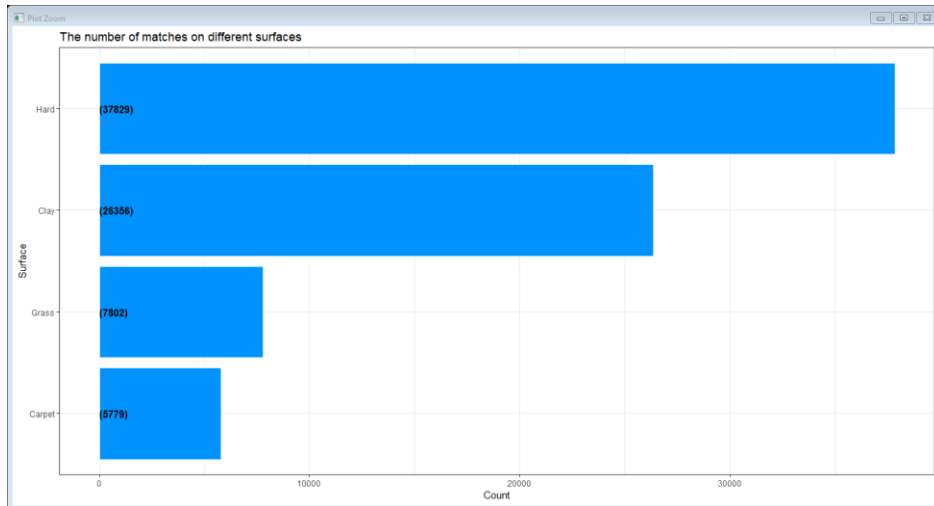


Fig 8. No of Matches on different surface

Fig 9, shows top 8 winners on all surfaces

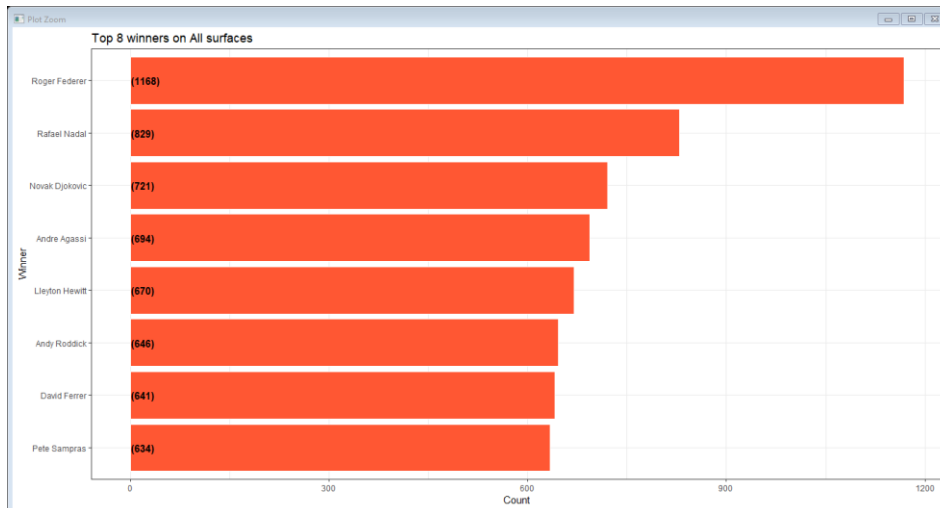


Fig 9. Top winners in all Surface

- **Tournament Levels**

Fig 10 shows the number of matches on different tournaments

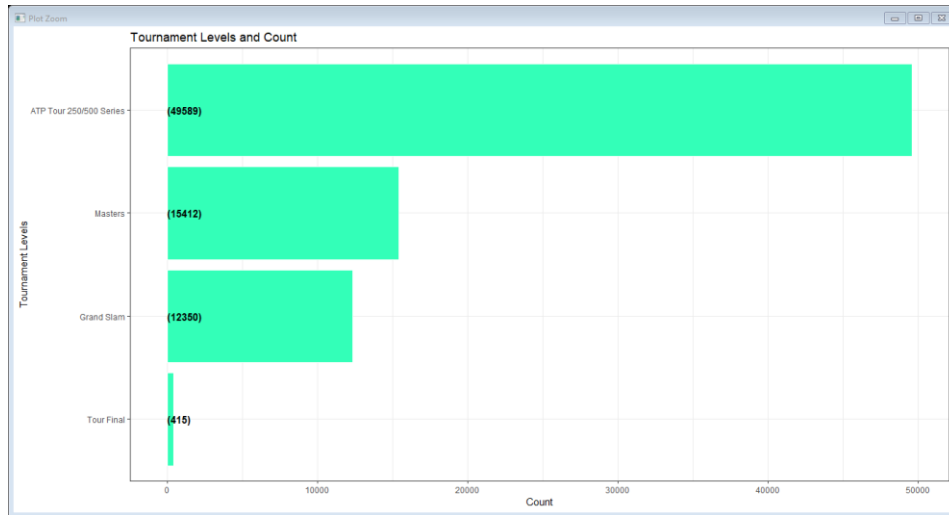


Fig 10. "Tournament Levels"

- Histogram Plot for all attributes is shown in Fig 11

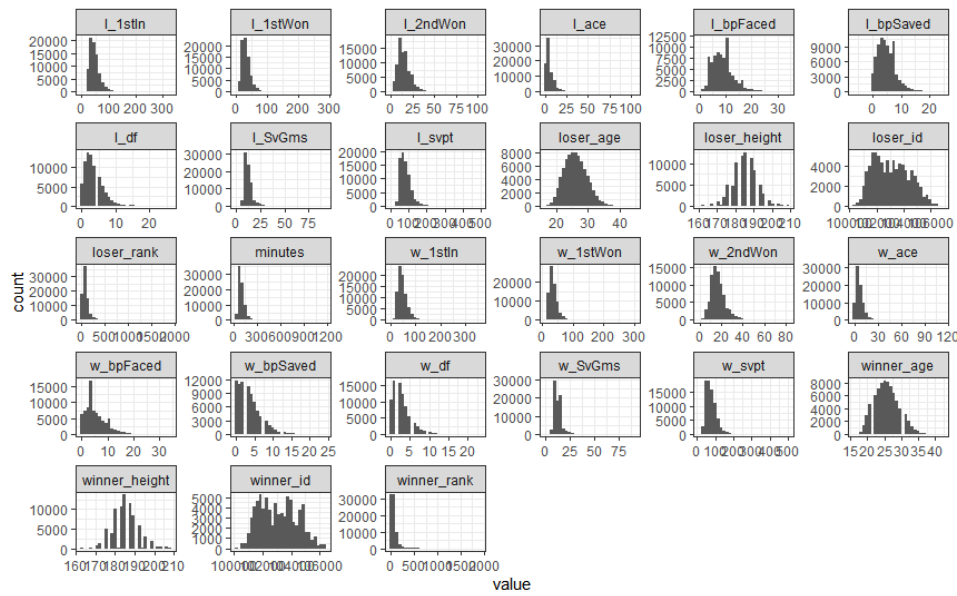


Fig 11. Histogram Plot for all features

- Density Plot for all attributes is shown in Fig 12

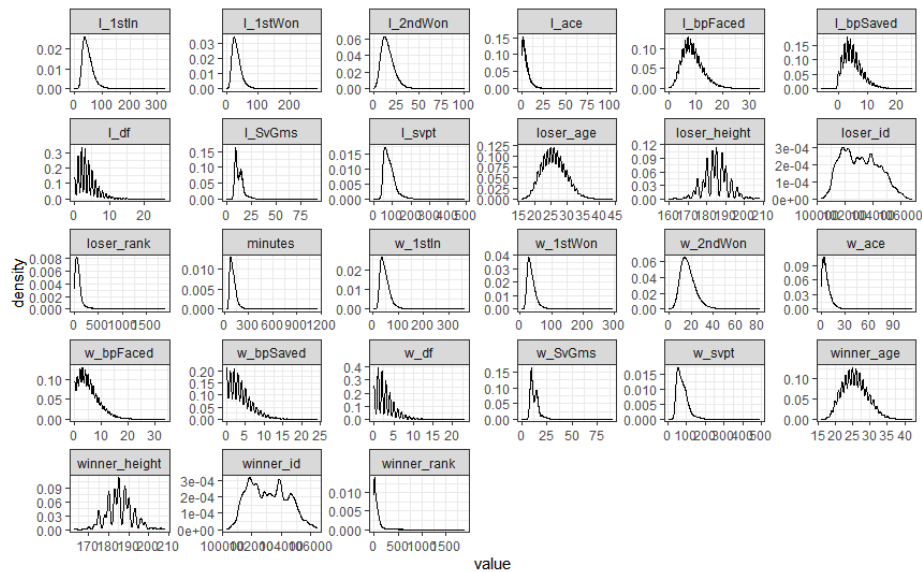


Fig 12. Density Plot for all features

## 5 Linear Regression Model

Regression analyzes are a statistical method very commonly used for creating a model of relation between two variables. One variable is called a predictor variable, the value of which is obtained through experimentation. The next variable is the answer variable, the value of which comes from the variable predictor.

These two variables in linear regression are associated by an equation in which the exponent (power) of the two variables is 1. A linear relation mathematically describes a line when drawn as a graph. a linear relationship A non-linear relationship where a variable's exponent is not equal to 1 produces a curve.

They are steps to follow to create a relationship, first, an experiment is conducted to obtain a comparison of measured values. Then, relation model constructed by using the `lm()` function in R. After that, mathematical equation using the coefficients from the generated model is found. Moreover, the relation model summary is getting to know the average prediction error (Residual). Finally, `predict()` function in R is used to predict the weight of new individuals.

In this section, a study is done on the effect of break points winning affects the first serve points won by the player.

### 5.1 Libraries used:

Some libraries used for linear regression model are: (*RDocumentation*, n.d.)

**Rvest**: make it easy to scrape data from HTML web pages.

**Tidyverse** Package: set of packages work for data representation and visualization, library tidyverse load the core **tidyverse** packages:

**Dplyr**: used for data manipulation.

**Tidyr**: help to create tidy data and describe a standard way of storing data.

**ggplot2**: dedicated to data visualization. It can greatly improve the quality of the graph.

**Purrr**: for functional programming

**MASS**: Functions and datasets to support Venables and Ripley

**Rcompanion**: to support summary and analysis in R

**Knitr**: to make flexible, fast dynamic report generation in R

**Carrrt**: is a series of functions to streamline the predictive modeling process.

## 5.2 Correlation

Use a Pearson correlation ( $r$ ), which calculates a linear association between two variables ( $x$  and  $y$ ). There are various approaches for the study of correlations. Also known as a parametric correlation test, it depends on how the data is distributed. It can only be used if the distribution of  $x$  and  $y$  is natural. The  $y = f(x)$  plot is called the linear return curve.

The variables used in this research are as follows: the dependent variable:

Dependent variable: *w\_1stWon* “Winner's point number of first Serve”

Independent variable: *w\_bpSaved* “Break Points Winning by Players”.

A correlation value 0.4518554 is established between the variables; a fair connection between two variables of Break Points that are won by the player and the number of games the player won. A positive (proportional) relation between the two variables implies that the higher the break points, the more points won. Fig 13 shows the result of correlation.

```
data: atp.data$w_bpSaved and atp.data$w_1stwon
t = 141.25, df = 77764, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.4462443 0.4574311
sample estimates:
      cor
0.4518554
```

Fig 13. Correlation Value between *1stWon* and *w\_bpSaved*

## 5.3 Simple Linear Regression Model

The final model has a modest Adjusted R-squared of 1.943. It is somewhat surprising to me that R-squared is that low, because I use a bunch of predictors that are only known after the game. Nevertheless, there is one predictor that is known before the game and it is difference in ranks. After finding the value of the linear regression, a plot has to be drawn to make the findings clearer. Fig 14 shows the linear regression value.

```
Coefficients:
(Intercept) atp.data$w_bpSaved
 28.700      1.943
```

Fig 14. Linear Regression Model Result

Fig 15 shows the plot that represent the linear regression result.

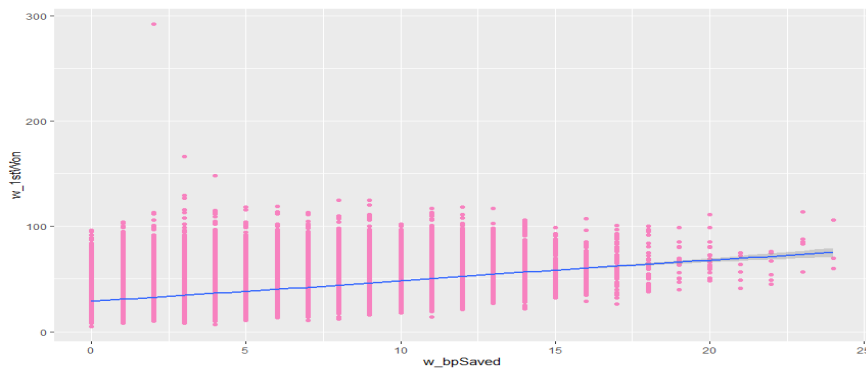


Fig 15. Linear Regression Plot

The predictive interval indicates the uncertainty around a single value, while the confidence interval indicates the uncertainty about the mean predictive value. Thus, the forecast interval will normally be much wider than the confidence interval for the same value.

Using a confidence interval when using a predictive interval will greatly underestimate the uncertainty in each predicted value (Bruce & Bruce, 2017).

The R code creates a scatter plot with lines: blue illustrates the regression line, grey illustrates the confidence. Red illustrates the prediction. Fig 16 illustrates the predict plot for the linear regression model.

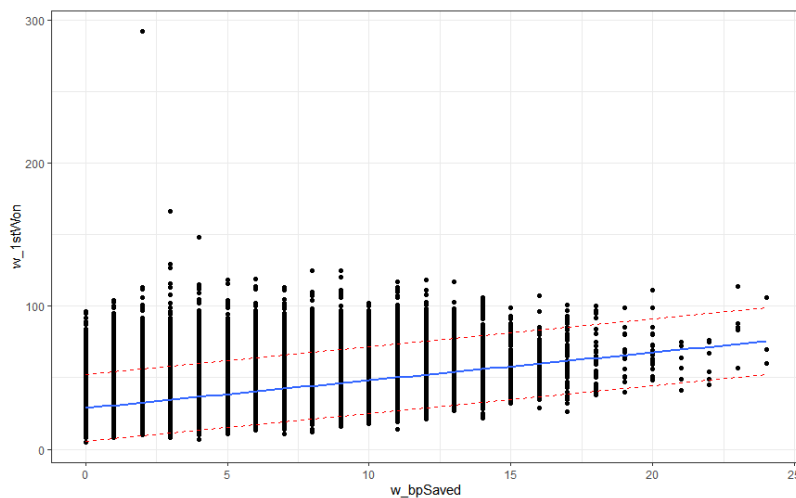


Fig 16. Predict Plot

## 5.4 Multiple Linear Regression

Several variables have recoded consisting of characters in numerical variables prior to the regression analysis (Surface, Rounder, Tourney Level and Tourney Name).

In addition to the correlations, the following examination will take the form of a linear several regressions to see additional relationships between variables. The relation between multiple explanatory variables and a dependent variable will be

analyzed. As the variable dependent, use the score (points) are used. Points is used as interdependent variable.

Before forming the multiple linear regression, there are some steps to follow to have clear findings. The first step is to measure the difference in the score as it shows the change between two tests. For the first test, the IT is determined by subtracting the value. Fig 17 shows the score before and after the score difference.

score	x
6-43-67-6(2)	1
6-33-67-6(6)	2

Fig 17. Score before score differences

The second step is to keep the only variables to be used in the analysis by using *select* function. Fig 18 shows the number of variables before and after using *select* function.

atp.data	77766 obs. of 45 variables
atp.data1	77766 obs. of 27 variables

Fig 18. Number of variables before and after *select()* function

The third step is to clean up some data to have clear results. For example any winner or loser hand hold unknown value is filled with 'U'. Then deal with winner's height loser's height and minutes null values and calculate the means of different groups of values. Alter that combine some variables to get more accurate result because the winner maybe a lose some times, according to that, the winner and loser age, the winner and loser height, winner and loser ace, winner and loser rank and winner and loser double faults are combined, then having a data frame by transforming all columns with columns that match a certain name.

Three plots are drawn to represent the relations between the variables. Fig 19(a) shows the relation of a player rank the winner and the player points. Although some points might differ a bit, it is globally very similar, no matter the surface played or the ranking of the player. Sense as higher in the ranking (low numbers) as you have more points (high numbers). This adverse relation is highlighted by a negative estimate of winner points. This relationship is seen in the graph and the tournament and round are considered. In addition, the top-ranking player does not normally play smaller tournaments like the ATP250 Series. This correlates to the facts. Fig 19(b) shows this relation and takes in consideration the Tourney Level and the Round. Low ranked player is playing first round of tournament while high ranked player, starts later in the tournament. Fig 19(c) highlights the relationship between Player Rank -symbolized by a number of points- and the points: the highest the points, the lower the rate.

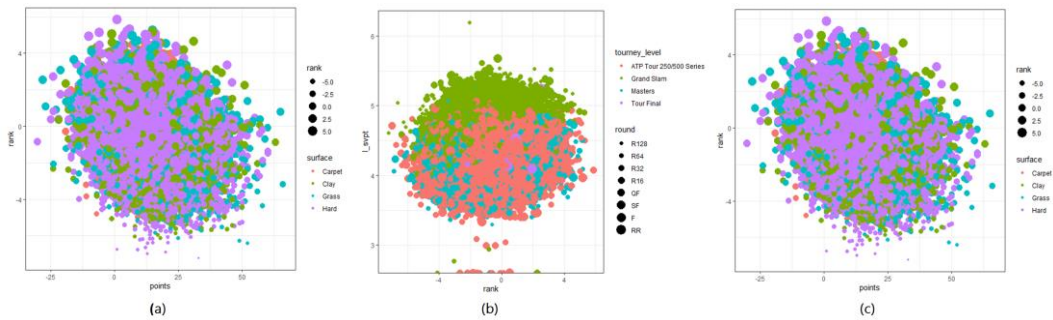


Fig 19. Visualization between player rank and player points, between Tourney level and the Round, between the player points player rank

We opted to consider many different numerical variables as explanatory variables and finally only maintained those with a sufficiently small probability value. One of the effects on the relationship of statistical studies is a probability value. The probability value is one of the effects on the relationship of statistical studies. It decides whether the estimate varies considerably from null and therefore whether the variable is significant in the model. Fig 20 shows a sample result of multiple linear regression correlation.

```
> atp.data1 %>%
+   keep(is.numeric) %>% cor(method = "pearson", use = "complete.obs")
minutes      minutes      w_svpt      w_1stIn      w_SvGms      w_bpFaced      l_svpt
w_svpt      0.898715678      1.000000000      0.81409924      0.862490306      0.61467933      NaN
w_1stIn      0.814099243      0.90241266      1.00000000      0.883728657      0.62036304      NaN
w_SvGms      0.862490306      0.90948026      0.88372866      1.00000000      0.54632769      NaN
w_bpFaced    0.614679327      0.68813144      0.62036304      0.546327691      1.00000000      NaN
l_svpt      NaN      NaN      NaN      NaN      NaN      NaN      1
l_1stIn      0.806083856      0.79860292      0.80020481      0.864599624      0.47954595      NaN
l_SvGms      0.865226931      0.89644735      0.87299751      0.983703351      0.54882665      NaN
l_bpFaced    0.396780837      0.30850914      0.31185872      0.357369046      0.37889223      NaN
points      -0.394150512      -0.42769474      -0.32982067      -0.311510869      -0.37881293      NaN
height      -0.028105555      -0.01982601      -0.03284373      -0.011681674      -0.02985456      NaN
age          0.006936869      0.01112146      0.01261154      0.009994563      0.00896747      NaN
rank         0.076075723      0.10198392      0.08513022      0.058247039      0.12110634      NaN
ace          -0.071141225      -0.04047651      -0.06621763      -0.025188776      -0.07613113      NaN
df           0.067104657      0.11301654      0.03548674      0.053721677      0.11938311      NaN
```

Fig 20. Sample Correlation of Multiple Linear regression

After plotting data in a graph, the model was created for different relationships. Three models of different relationships have been created that have different results.

Fig 21(a) shows the first model involving a different attribute: points, winner number of serve points, loser break point faced, the player height, the player rank, winner's number of serve games, loser's number of serve games, and player ace, using *poly()* function.

The *poly()* function in the stats package generates a matrix of (orthogonal) polyps over a set of values. When entering polynomial terms in a statistical model, the typical incentive is to ascertain whether the response is "curved."

A further important value is  $R^2$ , meaning that the percentage of variance that has been described is fewer than the variance of the dependent variable. A valid  $R^2$  should normally be more than 50 percent anywhere. As seen from the study that



very small  $R^2$  of 0.3777. The adjusted  $R^2$  is 0.3777 and the default error is 7250 on 10 with 77755 degrees. It is a very high-quality mistake.

In general, the adjusted  $R^2$  and the standard regression error should be examined. They are impartial estimators that correct the number of calculated coefficients and the sample size.

The standard mistake displays the average distance from the regression line for the values observed. It shows how wrong the regression model is with the solution variable units on average. The lower the rates, the closer they are to the line. Through this analysis, however, the model is not entirely confidential, even if we have significant estimates.

The second relation developed from the distance of Cook is used to identify influential outliers in regression analysis in a variety of predictor variables. This is a way of finding points that impact the regression model negatively. The metric is a combination of the heel and the residual value of each observation; the higher the heel and the more waste, the greater the distance of the cook.

The second relation is between winner rank, loser rank, surface, tourney name and round. As observed, that very small  $R^2$  of 0.3752. The  $R^2$  set is 0.3751 and the default error is 5837 on 8 with 77757 degrees. It is a very high-quality mistake. Fig 21(b) show the result of second relation in multiple linear regression.

Residuals:				
Min	1Q	Median	3Q	Max
-72.308	-4.219	-0.394	3.641	47.374

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.111e+02	9.330e-01	119.075	< 2e-16 ***
w_svpt	-1.743e+01	1.968e-01	-88.576	< 2e-16 ***
poly(l_bpFaced, 2)1	7.203e+02	7.554e+00	95.352	< 2e-16 ***
poly(l_bpFaced, 2)2	-6.851e+01	7.004e+00	-9.782	< 2e-16 ***
height	-4.473e-02	2.998e-03	-14.921	< 2e-16 ***
rank	-7.205e-01	1.769e-02	-40.720	< 2e-16 ***
df	-3.891e-01	8.363e-03	-46.523	< 2e-16 ***
poly(w_SvGms, 2)1	2.613e+03	4.161e-01	62.791	< 2e-16 ***
poly(w_SvGms, 2)2	3.487e+01	7.690e+00	4.535	5.77e-06 ***
l_SvGms	-1.885e+00	3.524e-02	-53.503	< 2e-16 ***
ace	2.921e-01	4.785e-03	61.057	< 2e-16 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.9 on 77755 degrees of freedom  
Multiple R-squared: 0.3777, Adjusted R-squared: 0.3777  
F-statistic: 4720 on 10 and 77755 DF, p-value: < 2.2e-16

(a)

```
lm(formula = points ~ w_svpt + l_bpFaced + rank + df + poly(w_SvGms, 2) + l_SvGms + ace, data = atp.data1)
```

Residuals:				
Min	1Q	Median	3Q	Max
-69.771	-4.230	-0.415	3.641	47.252

Coefficients:				
	Estimate	Std. Error	t value	Pr(> t )
(Intercept)	1.053e+02	9.329e-01	112.867	< 2e-16 ***
w_svpt	-1.739e+01	1.972e-01	-88.197	< 2e-16 ***
l_bpFaced	6.364e-01	6.642e-03	95.824	< 2e-16 ***
rank	-7.212e-01	1.773e-02	-40.675	< 2e-16 ***
df	-3.986e-01	8.356e-03	-47.705	< 2e-16 ***
poly(w_SvGms, 2)1	2.594e+03	4.164e+01	62.301	< 2e-16 ***
poly(w_SvGms, 2)2	3.155e+01	7.698e+00	4.098	4.17e-05 ***
l_SvGms	-1.882e+00	3.529e-02	-53.313	< 2e-16 ***
ace	2.625e-01	4.370e-03	60.057	< 2e-16 ***

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.914 on 77757 degrees of freedom  
Multiple R-squared: 0.3752, Adjusted R-squared: 0.3751  
F-statistic: 5837 on 8 and 77757 DF, p-value: < 2.2e-16

(b)

Fig 21. Linear Regression Model Result Relation1, Relation2

The third relation is between points with winner's number of serve points, lose's break points faced ,player's double faults , poly of winner's number of serve games, , loser's number of serve games winner and player ace. Through observation, very small  $R^2$  of .03751. The adjusted  $R^2$  is 0.375 and the default error is 5830 on 8 and 77702 degrees. It is a very high-quality relationship. Fig 22(a) show the result of third relation in multiple linear regression.

The fourth relation is between player points with winner's number of serve points, loser's break points faced, player's double faults, winner's number of serve games, loser's number of serve games winner and player ace. Through observation, the Residual standard error is 5.761 on 77703 degrees. Fig 22(b) shows the result of fourth relation in multiple linear regression.

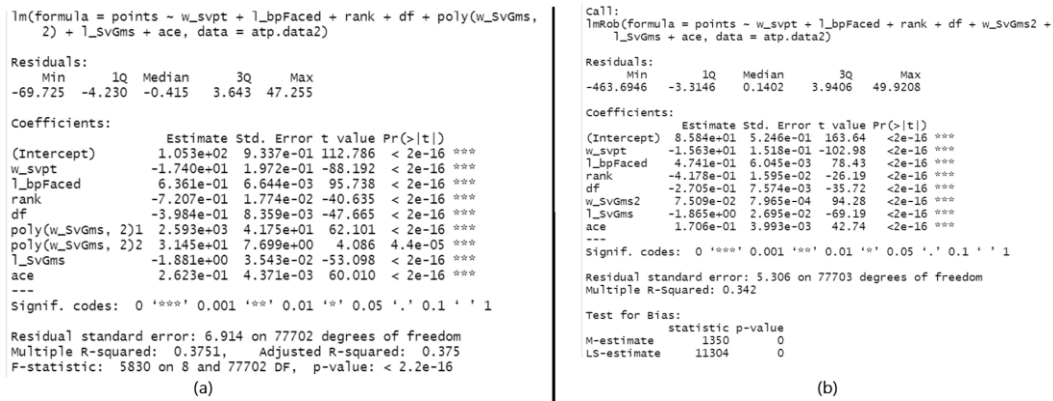


Fig 22. Multiple Linear Regression Model Result Relation3, Relation4

A moderate modified 0.342 R-squared standard is available in the final model. It is a little bit odd that R-squared is that weak, since we use a lot of game-only predictors. However, before the game is known a predictor and differential in ranks.

At the end the *predict* function implemented based on point greater than to twenty. Fig 23 illustrates a sample of prediction.

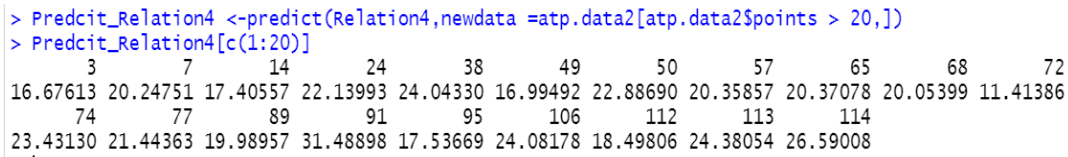


Fig 23. Prediction sample result for Relation 4

## 6 Decision Tree Model

For several factors, the decision tree model was selected to be in this project. Firstly, decision trees figured out that when compared with other models, it needed less time to prepare data during pre-processing. A "decision tree" does not require data normalization. There is no need for scaling data too. Moreover, missing values in the data often do not substantially impact the building decision tree process. A decision-making process is very straightforward for all strategic departments and stakeholders to understand (Khder et al., 2022) (K, 2019).

### 6.1 Libraries used:

In this section there are two prime packages used to build the decision tree which are Party and rpart packages with their dependencies: (*RDocumentation*, n.d.)

- **Using party package**
  - o **Library (grid):** a rewrite of the graphics layout abilities plus some support for interaction.
  - o **Library (mvtnorm):** computes multivariate normal and t probabilities, quantiles, random deviates and densities .
  - o **library(modeltools):** tools and classes for Statistical Models

- **Library (stats4):** Statistical Functions using S4 classes .
- **Library(party):** a laboratory for recursive partitioning
- **Library (partykit):** a Toolkit for recursive partitioning
- **Using rpart Package**
  - **Library(rpart):** recursive partitioning and regression trees
  - **Library (rpart.plot):** This functionality incorporates and expands the rpart kit "plot.rpart and text.rpart". These scales and updates the shown tree automatically .
  - **Library(RWeka):** R/Weka Interface

## 6.2 Choose the most appropriate features:

Feature selection is a method of selecting essential features to boost model efficiency while discarding those with irrelevant information (Bukhari et al., 2021). In this work we chose eight features to build the decision tree, which are surface, tourney level, round, winner rank, loser rank, best of, winner age, and loser age.

## 6.3 Check the structure of the selected features:

The surface as shown in Fig 24 is a character that means it needs to be converted to the factor because the features can be numerical (or integer), categorically unordered (i.e., factor), categorical ordered or censored. To obtain useful results from trees, choosing the suitable variable type in a data frame is essential.

```
> str(data.NeedForDT)
'data.frame': 77766 obs. of 8 variables:
 $ surface : chr "Hard" "Hard" "Hard" "Hard" ...
 $ tourney_level: Factor w/ 4 levels "ATP Tour 250/500 Series",...: 1 1 1 1 1 1 1 1 1 1 ...
 $ round : Factor w/ 8 levels "R128","R64","R32",...: 3 3 3 3 3 3 3 3 3 3 ...
 $ winner_rank : num 56 304 82 50 88 62 51 42 66 108 ...
 $ loser_rank : num 2 75 69 84 28 59 74 67 228 128 ...
 $ best_of : Factor w/ 2 levels "3","5": 1 1 1 1 1 1 1 1 1 1 ...
 $ winner_age : num 21 27 25 20 24 18 20 22 24 21 ...
 $ loser_age : num 23 25 23 21 20 24 19 25 20 20 ...
 - attr(*, "na.action")= 'omit' Named int 124 285 287 879 887 965 967 970 977 998 ...
 .. attr(*, "names")= chr "124" "285" "287" "879" ...
```

Fig 24. The structure of features needed in decision tree

## 6.4 Split the dataset into training and testing:

Train and test set aim to train the model in the train set and check the prediction in the test set. The common practice, depending on the research papers, divides the data 80-20, 80 percent serves as model training, and 20 percent for prediction. The test data set will not be touched until the model is ready. The number of observations for the training and testing set is shown in the Fig 25.





 data.test	15554 obs. of 8 variabl...	
 data.train	62212 obs. of 8 variabl...	

Fig 25. Number of observations

### 6.5 Select the target feature:

This step is to select the field from the data set that wants to predict. In this work the target variable is best of 5 or 3

### 6.6 Select the predictor features:

This step is to choose the variables from the data set that believe they "cause" changes in the value of the target variable. In this work, the selected predictors' variables are: surface, tourney level, round, winner rank, loser rank, winner age, and loser age.

### 6.7 Build the model using rpart package:

When running the model, the result shown in the Fig 26 shows that out of the seven variables, an essential variable for the prediction model is the tourney level in helping to classify the observations into two categories, which are the best of five and best of three. So if the tourney level is "ATP Tour 250/500 Series", "Masters," or "Tour Final," then it will go to the next node to check the round. If the round did not reach the final, then it would be categorized as the best of 3, so 51515 matches are not reached to the final round. On the other hand, if the match reached the final round, then it will go to recheck the tourney level. If the tourney level is "ATP Tour 250/500 Series" then it will be categorized to be best of 3 too, and there are 1134 out of 1272 matches in this category. Otherwise, if the tourney level is not "ATP Tour 250/500 Series" then it will go then surface node to check if it is hard then will check the tourney level again if its "Masters" then it will be categorized to be best of 3 matches and here we have 69 out of 102 matches are best of 3. Otherwise, if the matches, not "Master," then will be categorized as the best of 5, 11 out of 15 categorized in this category. Now, if the surface is not Hard, then there are 79 out of 94 matches' categories to be the best of 5. Else means the tourney level is "Grand" in this case, the matches will be categorized as best of 5, which are 9212 matches.

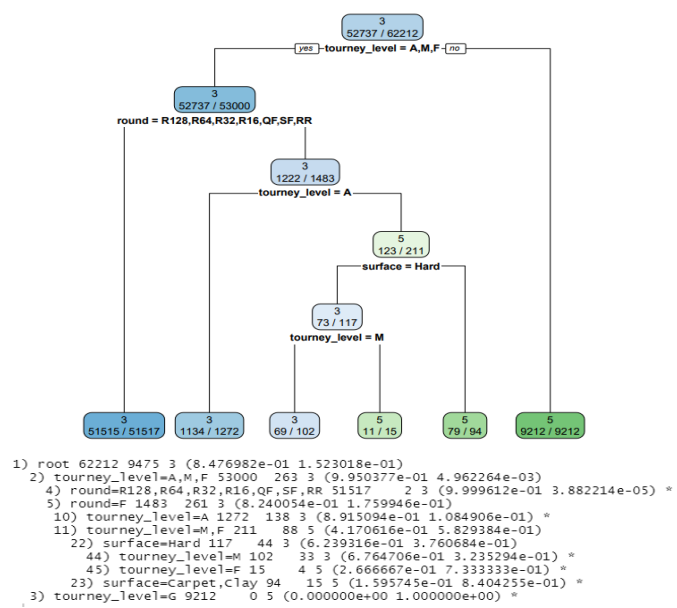


Fig 26. Training model result

## 6.8 Make a prediction using rpart package:

This step is to predict which matches are more likely to be the best of 5 or best of 3. It means they will know among those 15554 matches (test data set), which will best be 5 or 3, as shown in the Fig 27 sample of the predicted matches.

```
> predict_unseen_test <- predict(fit_Bestof, data.test, type = 'class')
> predict_unseen_test
64627 64628 64629 64630 64631 64632 64633 64634 64635 64636 64637 64638 64639 64640 64641 64642 64643 64644 64645
3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3
64646 64647 64648 64649 64650 64651 64652 64653 64654 64655 64656 64657 64658 64659 64660 64661 64662 64663 64664
3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3
64665 64666 64667 64668 64669 64670 64671 64672 64673 64674 64675 64676 64677 64678 64679 64680 64681 64682 64683
5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5
```

Fig 27. Prediction using rpart package

Create a table to count how many matches are classified as best of 5 and best of 3 and then compare to the correct classification, as shown in the Fig 28.

```
> table_mat_test <- table(data.test$best_of, predict_unseen_test)
> table_mat_test
  predict_unseen_test
  3      5
3 12383   31
5      2 3138
```

Fig 28. Count misclassified

The result shows that the model correctly predicted 12383 matches as best of 3 and classified 31 matches as best of 5. By analogy, the model misclassified 2 matches as best of 3 while they turned out to be best of 5

## 6.9 Measure performance *rpart* Package:

The accuracy measure can be computed with the confusion matrix for the classification task. The confusion matrix is a better option for classification success evaluation. The aim is to count the number of times exact instances are marked as false (Visa et al., 2011). Fig 29 shows the idea of confusion matrix in a simple way.

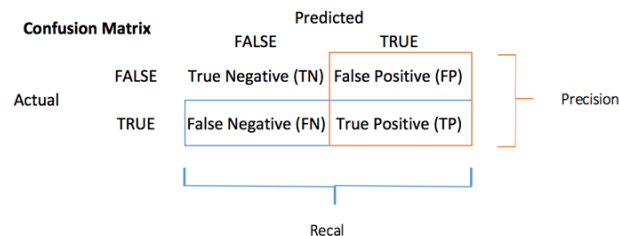


Fig 29. Confusion Matrix

Each row in a confusion matrix is the actual target, and each column is a predicted target. The first matrix row considers the best of three matches (false class): 12382 was classified as the best of three true negative (TN), whereas the

remainder was incorrectly classified as a best of five (false positive) (FP). The second row (positive class) finds the best 5 to be 3138 matches (true positives) (TP), while the false negative (FN) was 2 matches. The accuracy test from the confusion matrix can be computed as the following:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$accuracy = \frac{(3138 + 12382)}{(3138 + 12382 + 31 + 2)} = 0.997$$

Therefore, the performance of the model is 99.75% which is a good performance.

To ensure that our calculation is correct will obtain that by R using the following formula:

```
accuracy_Test <- sum(diag(table_mat)) / sum(table_mat)
```

Which is a proportion of true positive (TP) and true negative (TN) over the sum of the matrix.

The result of the model performance is shown in the Fig 30:

```
#-----
# measure the accuracy
#-----
accuracy_Test <- sum(diag(table_mat_test)) / sum(table_mat_test)
print(paste('Accuracy for test', round(accuracy_Test, digits = 3 )))
[1] "Accuracy for test 0.998"
```

Fig 30. Accuracy of testing model

## 6.10 Tune the hyper-parameters:

The decision tree has different parameters that control fit aspects. The parameters can be controlled with the *rpart.control* function in the *rpart library*. The primary purpose of tune the parameters is to try to improve the model over the previous value. This step will not affect too much in this case because our performance result is perfect since its note is less than 0.78. Anyway, we will tune the parameters until we get a better result than the previous one.

In the following explain the parameters that we want to tune:

- Build a function to return the accuracy
- Tune the max depth (in this case set 2)
- Tune the min number of samples a node must have before it can split (in this case set 200)
- Tune the min number of samples a leaf node must have (in this case set round (6/2))

In this work, the following parameters set gave us the best performance for the model as shown in Fig 31

```
minsplit = 200,
minbucket = round(6 / 2),
maxdepth = 2,
cp = 0)
```

Fig 31. Tune parameters in the model

In the Fig 32 shows the previous performance and new improved performance:

```
> print(paste('Accuracy for test', accuracy_Test*100))
[1] "Accuracy for test 99.7556898546998"
> print(paste('Accuracy for test', accuracy_tune*100))
[1] "Accuracy for test 99.9871415713"
```

Fig 32. Improved accuracy

### 6.11 Build the model using party package

Party package is another way to build, train and plotting the model using decision tree. In the following Fig 33 result shows the model using the party package. The result shown a big tree with many nodes, so the best way to get better result of the model is to prune the tree. Prune the tree is almost equivalent to the tune the parameters in rpart package.

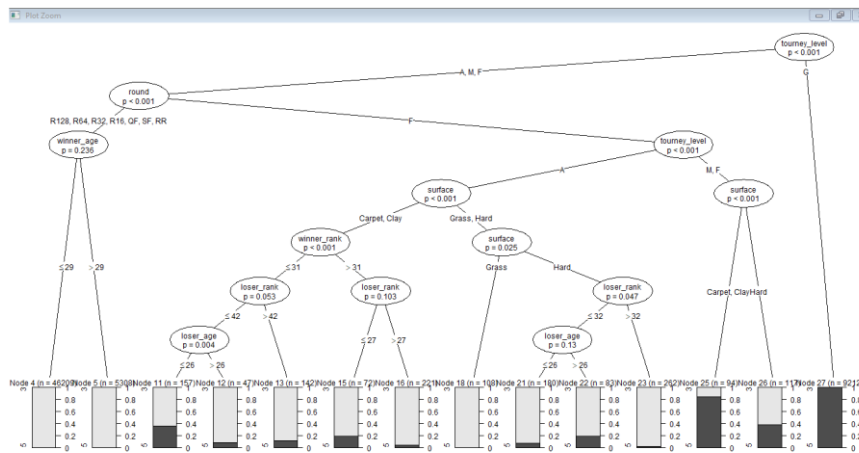


Fig 33. Decision tree using party package

After pruning the tree, the Fig 34 shows the model with fewer nodes and better results. When comparing this plotting with the one obtained by using the rpart package will find a slight difference.

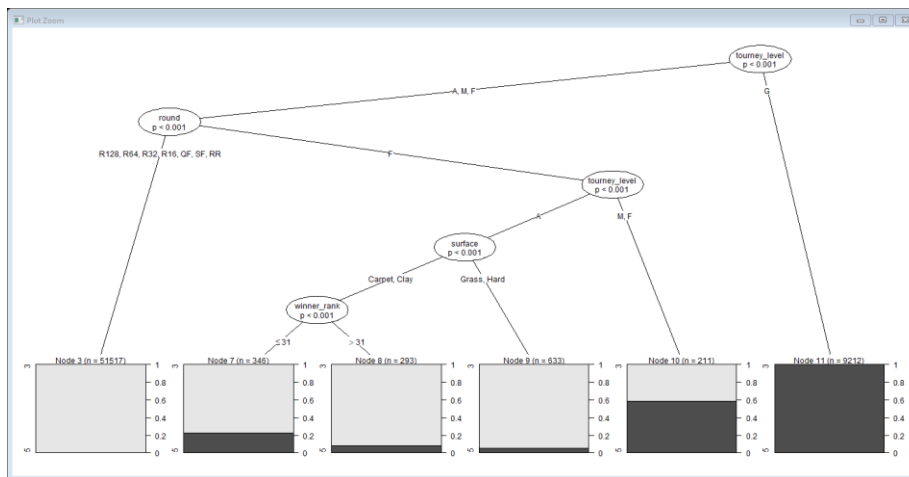


Fig 34. Prune tree using party package

### 6.12 Make a prediction of party package:

The used dataset have 15555 observations, so when predict by using type probability, can see that for all 15555 got two columns one for the best of 3 and



the other for best of 5 the first column shows the probability that the match will be best of 3 and the second column shows the probability that the match will be best of 5 as shown in Fig 35.

```
> predict_prob <-predict(fit_party, data.test, type = 'prob')
> predict_prob
      3      5
64627 0.9999612 3.882214e-05
64628 0.9999612 3.882214e-05
64629 0.9999612 3.882214e-05
64630 0.9999612 3.882214e-05
64631 0.9999612 3.882214e-05
64632 0.9999612 3.882214e-05
64633 0.9999612 3.882214e-05
64634 0.9999612 3.882214e-05
```

Fig 35. Sample of the predicted probability for each match

When predict by using response type then the result as shown in the Fig 36, the first two matches are predicted to be as best of 3. The third and fourth positions are predicted to be as best of 5 ... and so on.

```
> predict_response <-predict(fit_party, data.test, type = 'response')
> predict_response
64627 64628 64629 64630 64631 64632 64633 64634 64635 64636 64637 64638 64639 64640 64641 64642 64643 64644 64645
 3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3
64646 64647 64648 64649 64650 64651 64652 64653 64654 64655 64656 64657 64658 64659 64660 64661 64662 64663 64664
 3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3      3
64665 64666 64667 64668 64669 64670 64671 64672 64673 64674 64675 64676 64677 64678 64679 64680 64681 64682 64683
 5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5
64684 64685 64686 64687 64688 64689 64690 64691 64692 64693 64694 64695 64696 64697 64698 64699 64700 64701 64702
 5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5      5
```

Fig 36. Sample of the predicted response for each match

## 7. Conclusion and future work

This research focuses on the use of supervised machine learning on data from tennis players. The dataset includes information about tennis players, matches, and scores, among other things. Data preprocessing is also used to clean and remove abnormalities. The tennis huge data collection is analyzed using machine learning techniques: Linear regression and decision trees. The decision tree modeled the best of 3 or best of 5 sets of matches and predicted which set of matches would be considered best. Also, compare the matches to the correct classification. Linear regression analyzed the association between dependent and independent factors such as the number of first serves won by the winner as a dependent variable and break point winning by players as independent variables. Each model is implemented along-side a visualized graph of the obtained result.

Big data and big data analytics are currently the trendiest topics in the IT field(Khder, 2021). Now that new technologies are being adopted by data scientists, big data analysis has become easier than ever. Big data analytics is constantly evolving, so inevitably new models and techniques will emerge creating room for future work and further enhancement of this project. Potential ideas for future work implementations listed below:

- 1) Expand data visualization to obtain more knowledge
- 2) Implement other supervised machine learning models to enhance data classification and prediction
- 3) Implement unsupervised learning models such as K-means clustering
- 4) Apply deep learning for better knowledge discovery, application and prediction

## References



- [1] Bruce, P., & Bruce, A. (2017). Practical Statistics for Data Scientists. In *O Reilly*.
- [2] Bukhari, S. N. H., Jain, A., Haq, E., Khder, M. A., Neware, R., Bhola, J., & Lari Najafi, M. (2021). Machine Learning-Based Ensemble Model for Zika Virus T-Cell Epitope Prediction. *Journal of Healthcare Engineering, 2021*. <https://doi.org/10.1155/2021/9591670>
- [3] Burkart, N., & Huber, M. F. (2021). A survey on the explainability of supervised machine learning. In *Journal of Artificial Intelligence Research* (Vol. 70). <https://doi.org/10.1613/JAIR.1.12228>
- [4] Damanik, I. S., Windarto, A. P., Wanto, A., Poningsih, Andani, S. R., & Saputra, W. (2019). Decision Tree Optimization in C4.5 Algorithm Using Genetic Algorithm. *Journal of Physics: Conference Series, 1255*(1). <https://doi.org/10.1088/1742-6596/1255/1/012012>
- [5] Es-sabery, F., & Hair, A. (2019). A MapReduce C4.5 Decision Tree Algorithm Based on Fuzzy Rule-Based System. *Fuzzy Information and Engineering, 11*(4). <https://doi.org/10.1080/16168658.2020.1756099>
- [6] Fujo, S. W., Subramanian, S., & Khder, M. A. (2022). Customer churn prediction in telecommunication industry using deep learning. *Information Sciences Letters, 11*(1). <https://doi.org/10.18576/isl/110120>
- [7] Historical dictionary of tennis. (2012). *Choice Reviews Online, 49*(07). <https://doi.org/10.5860/choice.49-3634>
- [8] Jefferys, K. (2012). Historical Dictionary of Tennis. *The International Journal of the History of Sport, 29*(13). <https://doi.org/10.1080/09523367.2012.714285>
- [9] Johnson, D. (2022). *Decision Tree in R with Example*. <https://www.guru99.com/r-decision-trees.html>
- [10] K, D. (2019). *Top 5 advantages and disadvantages of Decision Tree Algorithm*. <https://dhirajkumarblog.medium.com/top-5-advantages-and-disadvantages-of-decision-tree-algorithm-428ebd199d9a>
- [11] Khder, M. A. (2021). Web scraping or web crawling: State of art, techniques, approaches and application. *International Journal of Advances in Soft Computing and Its Applications, 13*(3). <https://doi.org/10.15849/ijasca.211128.11>
- [12] Khder, M. A., Sayfi, M. A., & Fujo, S. W. (2022). Analysis of World Happiness Report Dataset Using Machine Learning Approaches. *International Journal of Advances in Soft Computing and Its Applications, 14*(1). <https://doi.org/10.15849/IJASCA.220328.02>
- [13] Lake, R. J. (2012). Historical Dictionary of Tennis. *Sport in History, 32*(4). <https://doi.org/10.1080/17460263.2012.746857>
- [14] Pant, A., & R. S. Rajput. (2019). Linear Regression Analysis Using R for Research and Development. In *Writing Qualitative Research Paper of International Standard* (pp. 180–195).
- [15] *RDocumentation*. (n.d.). Retrieved July 30, 2022, from <https://www.rdocumentation.org/>
- [16] Sackmann, J. (2020). *ATP Tennis Rankings, Results, and Stats*. [https://github.com/JeffSackmann/tennis\\_atp/archive/master.zip](https://github.com/JeffSackmann/tennis_atp/archive/master.zip)
- [17] Singh, A., Thakur, N., & Sharma, A. (2016). A review of supervised machine learning algorithms. *Proceedings of the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, INDIACom 2016*.

- [18] Sipko, M. (2015). Machine Learning for the Prediction of Professional Tennis Matches. *MEng Thesis*.
- [19] T Akinsola, J. E., Jet, A., & O, H. J. (2017). Supervised Machine Learning Algorithms: Classification and Comparison Machine Learning View project The Use Of BIG DATA in Mobile Analytics View project Supervised Machine Learning Algorithms: Classification and Comparison. *International Journal of Computer Trends and Technology*, 48(3).
- [20] Tour, A. (2022). *ATP rankings*. <https://www.atptour.com/en/rankings/singles>
- [21] Visa, S., Ramsay, B., Ralescu, A., & Van Der Knaap, E. (2011). Confusion matrix-based feature selection. *CEUR Workshop Proceedings*, 710.



**Moaiad Ahmad Khder** (Senior Member, IEEE) received the Ph.D. degree from the Faculty of Information Science and Technology, The National University of Malaysia, in 2015. He is currently an Assistant Professor with the Computer Science Department, Applied Science University, Bahrain. He has been working on the area of mobile environment, mobile database, data science, machine and deep learning, big data and cloud computing.



**Samah Wael Fujo**, received her Master in Computer Science and information technology from College of Information Technology, Ahlia University- Bahrain in 2021, and her Bachelor degree in Computer Science from Applied Science University – Bahrain in 2019. She is currently working in Nasser Artificial Intelligence Research & Development Center (NAIRDC) - Nasser Vocational Training Centre (NVTC) - Bahrain. She has been working on the area of data science, machine and deep learning.