Data Analysis Amount of Time Played Affects on levels and wins in Call of Duty Data with Algorithm DBSCAN and Algorithm Random Forest

Nurul Aini Lativah
Information System, Multimedia Nusantara University, Tangerang, Indonesia
nurul.lativah@student.umn.ac.id

Abstract - Call of Duty is one of first-person shooter video game franchise published by Activision. The series originally focused on the World War II setting. First Person Shooter (FPS) is a subgenre of shooter video games centered on gun and other weapon-based combat in a first-person perspective and controlling the player character in a three dimensional space. Since there are so many things that could give impact for player can win the game, one of the aspect is amount of timePlayed that the author wants to analyze on levels and wins in Call of Duty Data.

Index Terms - Call of Duty, First Person Shooter Games, TimePlayed, Random Forest, DBSCAN

I. INTRODUCTION

A. Background of the study

Data is a raw fact that has not been Processed. Every day humans continue to produce data so that along with the times. Therefore, the term Big Data appears in society. Big Data points to technologies and initiatives that involve data that are so diverse, rapidly changing, or super-large that it is too difficult for conventional technology, expertise, and infrastructure to be able to handle them effectively. For researchers and business people, Big Data or big data can be used to produce a pattern or form that produces a New knowledge [1]. Therefore, Big Data is seen as an asset or resource that can be utilized to gain new knowledge and insights. Big Data has a variety of data, with formats and types of data that are very diverse so that it requires a special process to be able to process it[2]. Big Data must be able to process large amounts of data in a short time so that data can be useful not only because of the information generated but also because of its speed to process data into current information. Veracity is related to data certainty and value related to the value of the benefits of the information generated[3]. In today's modern era, not a few people play games. Games can be a hobby or a refreshing for people. From young children to adults now they can play games. Starting from digital and nondigital games. One of the digital games that will be discussed in this study is the Call Of Duty game. Call of Duty is a first-person shooter video game published by Activision. This game focuses on a game set in World War II. Call Of Duty was started in 2003 by Infinity Ward.

TimePlayed is the time that player spent to play a game. TimePlayed is used to count the time in one game so the player know how much the time they have spent on a game. So, in this study, the author uses the target variable timePlayed from the dataset to analyze how amount of Time Played affects on levels and wins in Call of Duty Data with several data mining methods, such as random forest, and DBSCAN

B. Problem

Call of Duty is one of first-person shooter video game franchise published by Activision. The series originally focused on the World War II setting. First Person Shooter (FPS) is a sub-genre of shooter video games centered on gun and other weapon-based combat in a first-person perspective, with the player experiencing the action through the eyes of the protagonist and controlling the player character in a three dimensional space. They are often played competitively online. Many aspects of the game that can affect the probability of winning in this first person shooter game, such as map knowledge, comfortable role or gun, basic moves, and much more. Since there are so many things that could give impact for player can win the game, one of the aspect is amount of timePlayed that I want to analyze on levels and wins in Call of Duty Data.

II. LITERATURE REVIEW

A. Data Mining

Data mining is a technique that allows to obtain patterns or models from gathered data. This technique is applied in all kind of environments such as the biological field, educational and financial applications, industry, police and political process. Within data mining, there are several techniques, among which are naïve bayes, decision tree, logistic regression, random forest and support vector machine[4].

B. Random Forest

The Random Forest is made up of several decision trees, each decision tree will be full growth, it do not need to cut processing, the more tree it has the more accurate the result will be, and it will not over fitting. The random forest algorithm will do the overall estimate, and it has the advantage of automatic feature selection etc. So we have the following main problems to be solved[5]. *Random Forest* means the development of *the CART* method, using *the bootstrap aggregating (bagging)* method and random *feature selection*[6].

Random Forest grows multiple decision trees which are merged together for a more accurate prediction. The logic behind the Random Forest model is that multiple uncorrelated models (the individual decision trees) perform much better as a group than they do alone. When using Random Forest for classification, each tree gives a classification or a “vote.”

The forest chooses the classification with the majority of the "votes." When using Random Forest for regression, the forest picks the average of the outputs of all trees. The key here lies in the fact that there is low (or no) correlation between the individual models—that is, between the decision trees that make up the larger Random Forest model. While individual decision trees may produce errors, the majority of the group will be correct, thus moving the overall outcome in the right direction[7].

C. DBSCAN

The DBSCAN is a base algorithm of density based clustering. It requires user to specify two global input parameters i.e. MinPts and Eps. The density of an object is the number of objects in its Eps-neighborhood of that object. DBSCAN does not specify the upper limit of a core object. So due to this, the clusters detected by it, are having wide variation in local density and forms clusters of any arbitrary shape. DBSCAN starts with an arbitrary point p and retrieves all points' density reachable points from p wrt. Eps and MinPts. If p is a core point, this procedure yields a cluster wrt. Eps and MinPts. If p is a border point, no points are density reachable from p and DBSCAN visits the next point of the database[8]. DBSCAN uses this concept of density to cluster the dataset. Now to understand the DBSCAN algorithm clearly, we need to know some important parameters. Important parameters of the DBSCAN algorithm is epsilon, Neighbourhood, min_sample. Now based on these two parameters i.e., epsilon and min_samples, we are first going to classify every point in our dataset into three categories. They are core points, Boundary points or border points, and noise points[9].

III. METHODOLOGY

A. Object of Research

This Research focuses on processing Call of Duty Player data which the dataset consists of 19 variables and 1558 observations. Among 19 variables, I selected 9 variables out of 19 that I want to analyze, which are wins, level, kills, deaths, averageTime, headshots, misses, shots, and timePlayed. As for the target variable, I focused on timePlayed and used Rstudio as the tool for analyzing the data.

B. Method of Collecting Data

The data the author uses for this research is not primary data because it does not come from survey results or other data collection methods that require researchers to be involved in the actual data collection process. The data used in this study is secondary data, namely data that has been collected by previous parties. Primary data collection cannot be done on the basis of a consideration, namely the limited reach and also the ability to collect sales data of certain companies independently in a relatively short period of time. The collected secondary data is downloaded from the kaggle site. Kaggle is a site / platform that holds competitions in the field of Data Science. In addition, this site is also one of the common sources of Data Science learning. Therefore, to support kaggle researchers provide a variety of datasets with various variations.

C. Research Method

In the process of processing data will be done using R. R is a programming language as well as a computing program used to support statistical analysis and graphing activities. R is related to Rstudio. RStudio is the Integrated Development Environment (IDE) for R[10]. In addition, before the data processing process is carried out, the data must be validated first to ensure the correctness and certainty of the data to be used in the research. R also facilitates the function. This devaluing. The research dataset containing the ability to play Call Of Duty will be studied and analyzed using two algorithms, namely the DBSCAN algorithm and the Random Forest algorithm. Both of these algorithms fall within the scope of data mining. In data mining has many techniques including classification, clustering, association, regression, forecasting, sequence analysis, and deviation analysis. But in this study will use clustering and classification techniques. Thus, DBSCAN was selected to conduct clustering analysis and Random Forest algorithm to perform classification analysis of datasets.

Here is the stage of data analysis conducted by researchers using the DBSCAN algorithm[11]:

1. Classify the points.
2. Discard noise.
3. Assign cluster to a core point.
4. Color all the density connected points of a core point.
5. Color boundary points according to the nearest core point.

Next is the application of random forest algorithm to perform classification techniques. Classification is the act of giving a group to every circumstance. Each state contains a group of attributes, one of which is a class attribute[12]. This method is necessary to find a model that can describe that attribute class as a function of the inputattribute. Here is the stage of data analysis conducted by researchers using the Random Forest algorithm[13]:

1). The algorithm select random samples from the dataset provided.

2). The algorithm will create a decision tree for each sample selected. Then it will get a prediction result from each decision tree created.

3). Voting will then be performed for every predicted result. For a classification problem, it will use mode, and for a regression problem, it will use mean.

4). And finally, the algorithm will select the most voted prediction result as the final prediction.

IV. RESULTS AND DISCUSSION

A. Data Validation

Data validity is a series of forms of precision over degrees in research variables that connect the research process on the research object with the data reported by a researcher[14]. The benefit of data validation is that it improves accuracy because it reduces errors in research data[15].

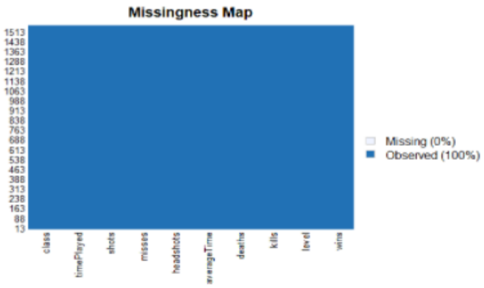
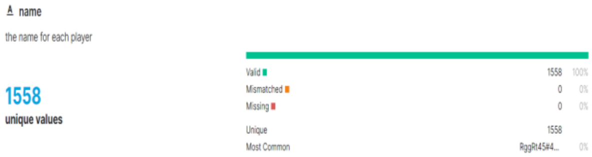


Figure 1. MissMap Data



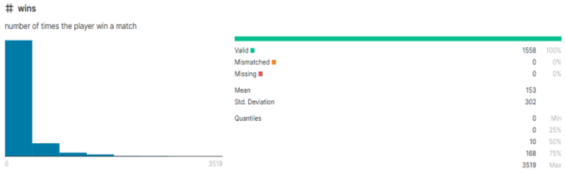
name

the name for each player

1558
unique values

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Unique	1558	
Most Common	RggRt45#4...	0%

Figure 2. Details of Validity of name

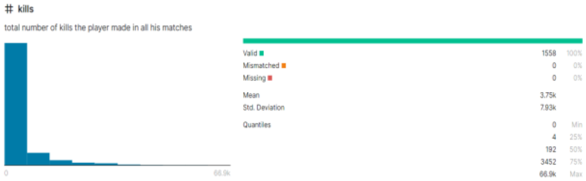


wins

number of times the player win a match

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	153	
Std. Deviation	302	
Quantiles	0	Min
	0	25%
	10	50%
	168	75%
	3519	Max

Figure 3. Detail of Validity of wins

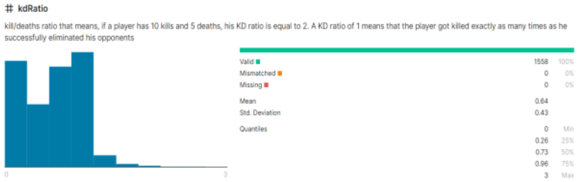


kills

total number of kills the player made in all his matches

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	3.75k	
Std. Deviation	7.93k	
Quantiles	0	Min
	4	25%
	192	50%
	3452	75%
	66.9k	Max

Figure 4. Details of Validity of kills

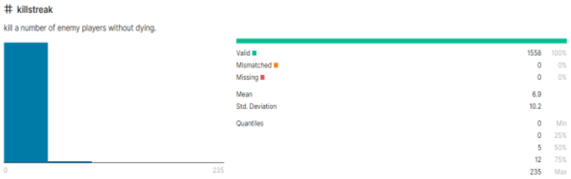


kdRatio

kill/deaths ratio that means, if a player has 10 kills and 5 deaths, his KD ratio is equal to 2. A KD ratio of 1 means that the player got killed exactly as many times as he successfully eliminated his opponents

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	0.64	
Std. Deviation	0.43	
Quantiles	0	Min
	0.26	25%
	0.73	50%
	0.96	75%
	3	Max

Figure 5. Details of Validity of kdRatio

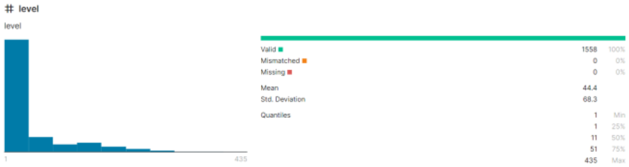


killstreak

kill a number of enemy players without dying.

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	6.9	
Std. Deviation	10.2	
Quantiles	0	Min
	0	25%
	5	50%
	12	75%
	235	Max

Figure 6. Details of Validity of killstreak

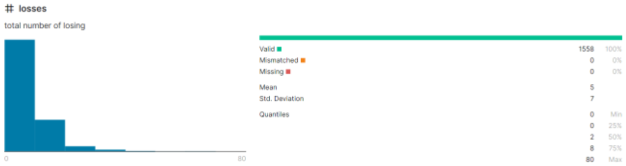


level

level

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	44.4	
Std. Deviation	68.3	
Quantiles	1	Min
	1	25%
	11	50%
	51	75%
	435	Max

Figure 7. Details of Validity of Level

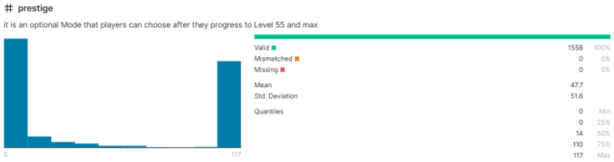


losses

total number of losing

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	5	
Std. Deviation	7	
Quantiles	0	Min
	0	25%
	2	50%
	8	75%
	80	Max

Figure 8. Details of Validity of Losses

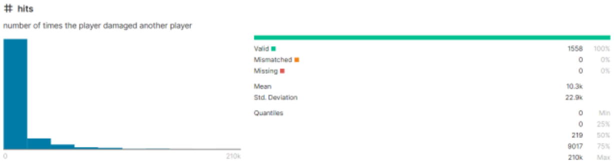


prestige

it is an optional Mode that players can choose after they progress to Level 55 and max

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	47.7	
Std. Deviation	51.6	
Quantiles	0	Min
	0	25%
	14	50%
	110	75%
	117	Max

Figure 9. Details of Validity of prestige

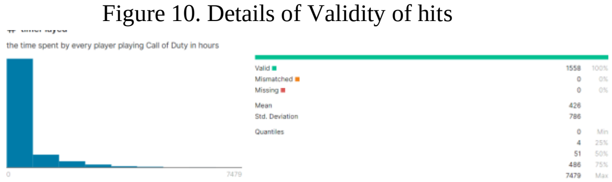


hits

number of times the player damaged another player

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	10.3k	
Std. Deviation	22.9k	
Quantiles	0	Min
	0	25%
	219	50%
	9017	75%
	210k	Max

Figure 10. Details of Validity of hits



the time spent by every player playing Call of Duty in hours

Valid 🟩	1558	100%
Mismatched 🟧	0	0%
Missing 🟥	0	0%
Mean	426	
Std. Deviation	786	
Quantiles	0	Min
	4	25%
	51	50%
	486	75%
	7479	Max

Figure 11. Details of Validity Time played

B. Data Visualization

Data visualization is used to make it easier for data readers to understand the results of processed data. Especially when the data to be processed is very much. The processing of very large amounts of data is known as data *minning*[16].

1) Numerical Data Visualization Using Scatter Plot

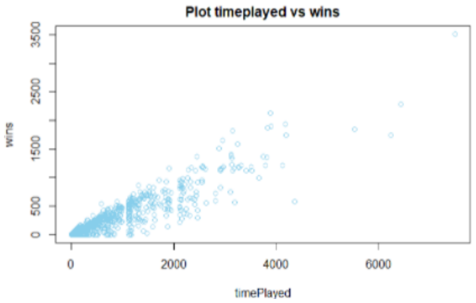


Figure 12. Numerical Data Visualization Using Scatter Plot

2) Numerical Data Visualization Using Histogram

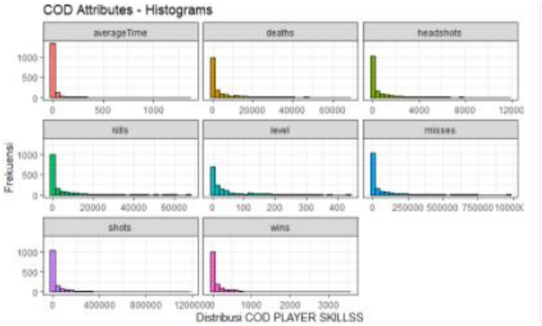


Figure 13. Visualization Variables Using Boxplot

3) Visualization Variables Using Boxplot

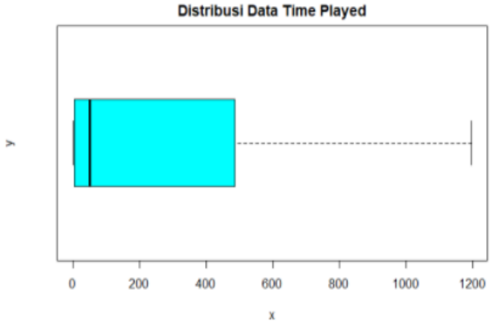


Figure 14. Visualization Variables Using Boxplot

4) Visualization Data Using ggpairs

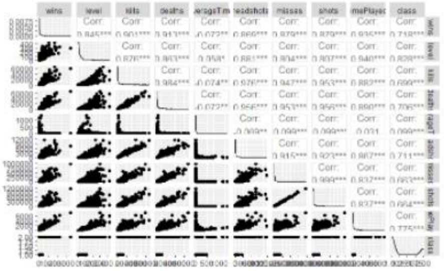


Figure 15. Visualization Data Using ggpairs

5) Numerical Data Visualization Using Density

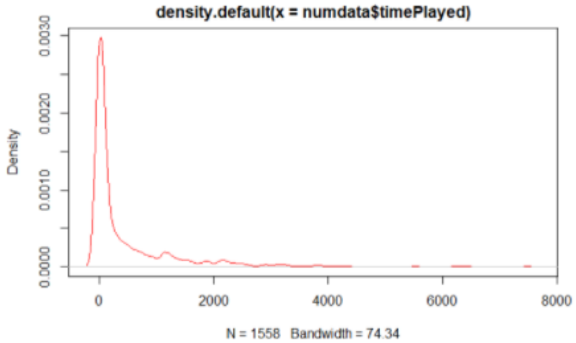


Figure 16. Numerical Data Visualization Using Density

C. Application of the Random Forest

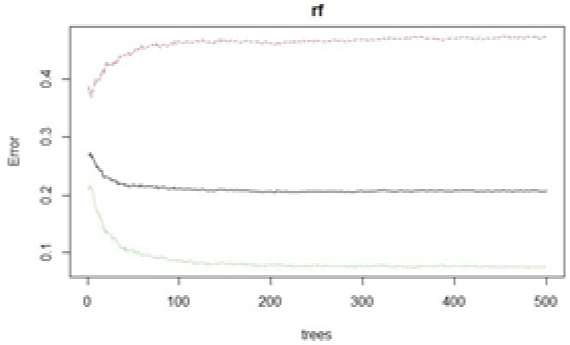


Figure 17. Random Forest

The lot above explains that there are three parts in the plot above. There are red, black and green colors. The plot above is a visualization of checking how many *trees* have *erorr* data.

D. Application of the DBSCAN Algorithm

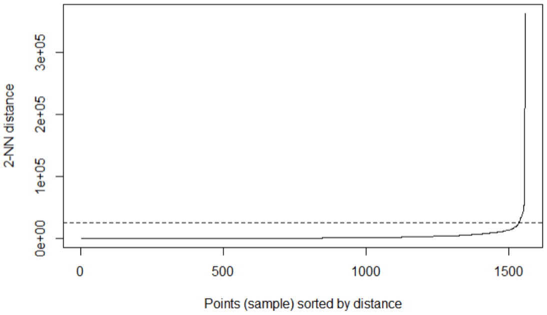


Figure 18. 2NN distance

Through the picture above the researchers chose z = 2. having a distance array and the ith entry in the array it will represent the2nd neighbor's distance from the i-point data point. And then the researchers will sort this array of distances and the researchers will plot them like this. On the y-axis, the researcher will only have the distance and on the x-axis, the researcher will have the index (i). because the index will increase the distanceof the2nd data point from that point will also increase. because the index will increase the distanceof the2nd data point from that point will also increase.

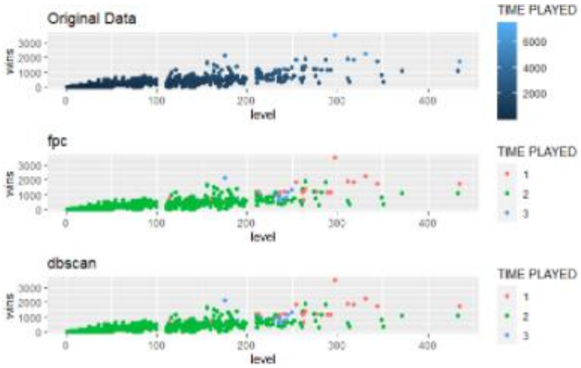


Figure 19. DBSCAN

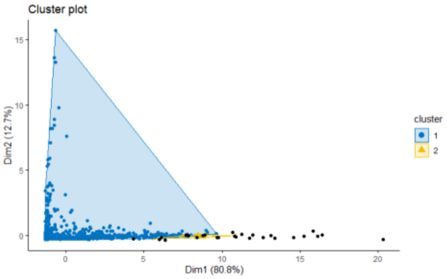
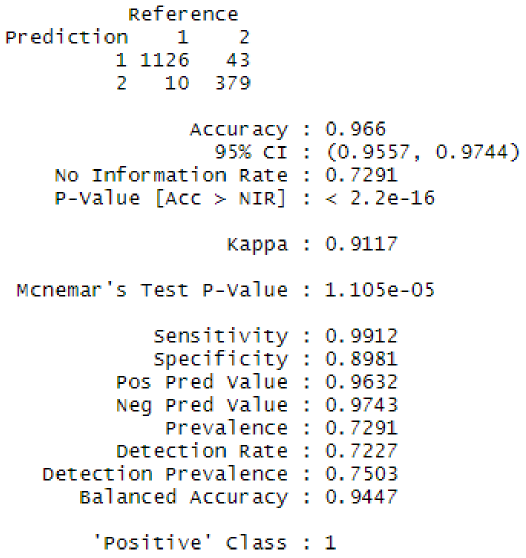


Figure 20. DBSCAN

The division of clusters is generally divided into 2 clusters. It is seen from the plot above that cluster 1 is more mirrored compared to cluster 2 which looks very little. The result of calculating the accuracy of this DBSCAN model that has been made is 77.09%.

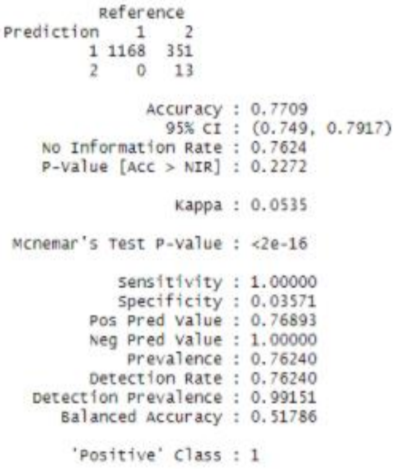
F. Comparison of the Model

From the level of accuracy, it will be seen whether the model can be used to make predictions or not. An accuracy rate above 50% indicates that the model can be used with a higher degree than random guessing. A higher level of accuracy than others will indicate which model is better to use in real life predictions.



```
          Reference
Prediction    1    2
         1 1126   43
         2   10  379
                                          
               Accuracy : 0.966           
                 95% CI : (0.9557, 0.9744)
    No Information Rate : 0.7291          
    P-Value [Acc > NIR] : < 2.2e-16       
                                          
                  Kappa : 0.9117          
                                          
 Mcnemar's Test P-Value : 1.105e-05       
                                          
            Sensitivity : 0.9912          
            Specificity : 0.8981          
         Pos Pred Value : 0.9632          
         Neg Pred Value : 0.9743          
             Prevalence : 0.7291          
         Detection Rate : 0.7227          
   Detection Prevalence : 0.7503          
      Balanced Accuracy : 0.9447          
                                          
       'Positive' Class : 1               
```

Figure 21. Random Forest Accuracy



```
           Reference
Prediction    1    2
         1 1168  351
         2    0   13

               Accuracy : 0.7709
                 95% CI : (0.749, 0.7917)
    No Information Rate : 0.7624
    P-Value [Acc > NIR] : 0.2272

                  Kappa : 0.0535

 Mcnemar's Test P-Value : <2e-16

            Sensitivity : 1.00000
            Specificity : 0.03571
         Pos Pred Value : 0.76893
         Neg Pred Value : 1.00000
             Prevalence : 0.76240
         Detection Rate : 0.76240
   Detection Prevalence : 0.99151
      Balanced Accuracy : 0.51786

       'Positive' Class : 1
```

Figure 22. DBSCAN Accuracy

V. Conclusion

In general, based on data analysis activities the ability of Call Of Duty game users using two algorithms that have been spelled out in the results analysis section, it can be concluded that random forest algorithms are better to be used in analyzing this data because of its very high accuracy. The results of the accuracy calculations of each algorithm used showed that the output accuracy value of the DBSCAN algorithm, which is 77.09% is much lower when compared to the output accuracy value of the Random Forest algorithm model of 96.60%. Here is the summary of the Random Forest and DBSCAN algorithms.

1) Random Forest

Confusion Matrix	**Random Forest**
Accuracy	0.966
Kappa	0.9117
95%CI	(0.9557, 0.9744)

Table 1. Comparison Random Forest

2) DBSCAN

Confusion Matrix	**DBSCAN**
Acurracy	0.7709
Kappa	0.0535
95%CI	(0.749, 0.7917)

Table 2. Comparison DBSCAN

Game players or companies in the field of gaming competitions, can take advantage of the results of this study to find out the grouping of one's abilities in playing games. In addition, it can learn abilities with time that affects with improvement or nothing. That way, it will be seen the extent to which the game player has the ability. However, keep in mind that this research is only limited in the scope of playerkills data analysis, then it may be possible to do using other data that is more diverse.

REFERENCES

[1] Natasuwarna, A. P. (2019). *Tantangan Menghadapi Era Revolusi 4.0 - Big Data dan Data Mining. Seminar Nasional Hasil Pengabdian Kepada Masyarakat 2019, 23-27.*

[2] Kusumasari, D., & Rafizan, O. (2017). *Studi Implementasi Sistem Big Data untuk Mendukung Kebijakan Komunikasi dan Informatika.* Jurnal Masyarakat Telematika dan Informasi, 8(2), 81-96.

[3] Sirait, E. R. (2016). *Implementasi Teknologi Big Data di Lembaga Pemerintahan Indonesia.* Jurnal Penelitian Pos dan Informatika, 6(2), 113-136.

[4] Mrs. Bharati M. Ramageri, *"DATA MINING TECHNIQUES AND APPLICATIONS,"* CiteSeer, vol. 1, pp. 301-305, 2016.

[5] WEIWEI LIN, ZIMING WU, LONGXIN LIN, ANGZHAN WEN, JIN LI, *"An Ensemble Random Forest Algorithm for Insurance Big Data Analysis,"* IEEE Access, vol. 5, 2017.

[6] Breiman L. (2001). *Random Forests*. Machine Learning.

[7] Rachel M. (2021). *What Is Random Forest.*

[8] Manisha Naik Gaonkar & Kedar Sawant, *"AutoEpsDBSCAN : DBSCAN with Eps Automatic for Large Dataset," ACADEMIA*, vol. 2, no. 2, pp. 11-16, 2018.

[9] Shree D. (2021). *Understand The DBSCAN Clustering Algorithm.* AnalyticVidhya.

[10] Amanda Pratama P. (2018). *Belajar Data Science : Langkah Awal Mengenal R dan Rstudio.*

[11] Vijjini M. (2020). *How to Use DBSCAN Effectively*. Toward data science.

[12] Desi S.L. (2016). *Pengertian Classification, Clustering, As Sociation, Regression, Fore Casting, Sequence Analysis, Deviation Analysis.*

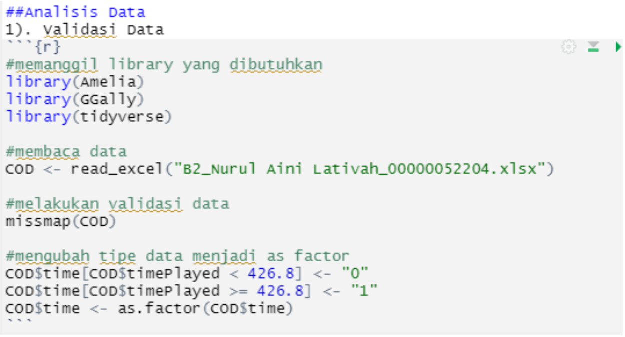
[13] Davis David. (2020). *Random Forest Classifier Tutorial: How to Use Tree-Based Algorithms for Machine Learning*. Machine Learning.

[14] Sugiyono. (2016). *Metode Penelitian Kuantitatif Kualitatif dan R&D.* Alfabeta: Bandung

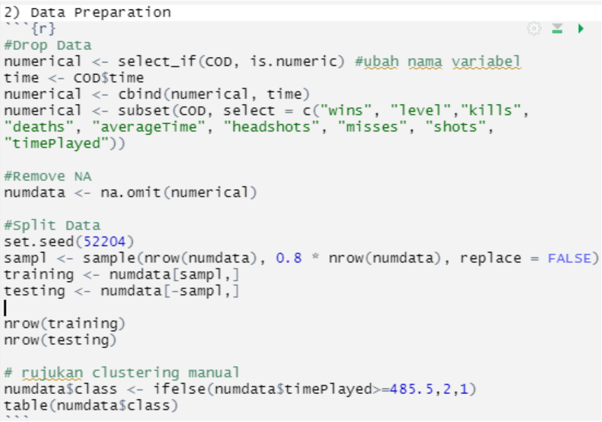
[15] Rina Hayati. (2020*). Pengertian Validasi Data Penelitian, Metode, dan Contohnya.*

[16] Windi Irmayani. (2021). *Visualisasi Data Pada Data Mining Menggunakan Metode Klasifikasi Naïve Bayes. Jurnal Khatulistiwa Informatika.*

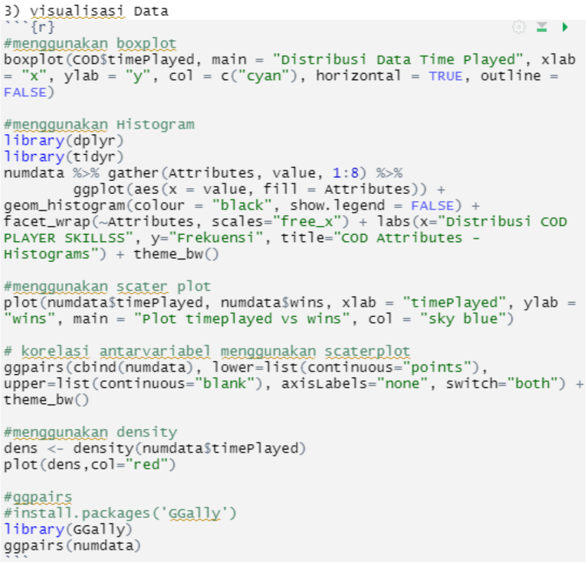
LAMPIRAN



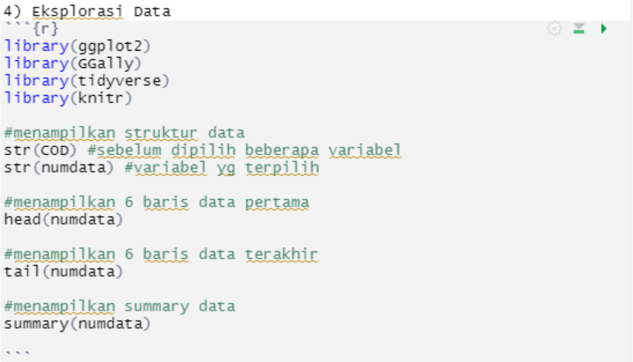
~~~
##Analisis Data
1). Validasi Data
```{r}
#memanggil library yang dibutuhkan
library(Amelia)
library(GGally)
library(tidyverse)

#membaca data
COD <- read_excel("B2_Nurul Aini Lativah_00000052204.xlsx")

#melakukan validasi data
missmap(COD)

#mengubah tipe data menjadi as factor
COD$time[COD$timePlayed < 426.8] <- "0"
COD$time[COD$timePlayed >= 426.8] <- "1"
COD$time <- as.factor(COD$time)
```
~~~



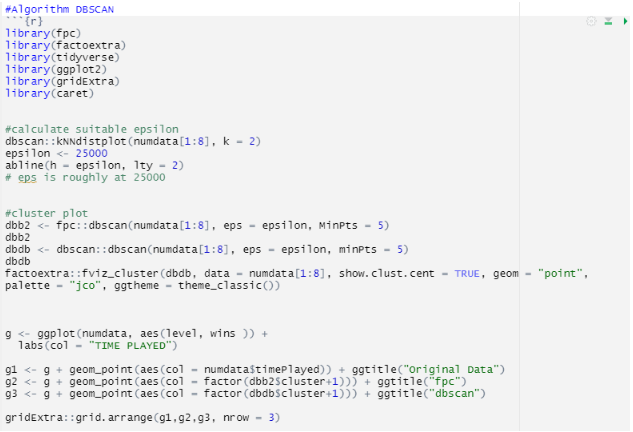
~~~
2) Data Preparation
```{r}
#Drop Data
numerical <- select_if(COD, is.numeric) #ubah nama variabel
time <- COD$time
numerical <- cbind(numerical, time)
numerical <- subset(COD, select = c("wins", "level","kills",
"deaths", "averageTime", "headshots", "misses", "shots",
"timePlayed"))

#Remove NA
numdata <- na.omit(numerical)

#Split Data
set.seed(52204)
sampl <- sample(nrow(numdata), 0.8 * nrow(numdata), replace = FALSE)
training <- numdata[sampl,]
testing <- numdata[-sampl,]

nrow(training)
nrow(testing)

# rujukan clustering manual
numdata$class <- ifelse(numdata$timePlayed>=485.5,2,1)
table(numdata$class)
```
~~~



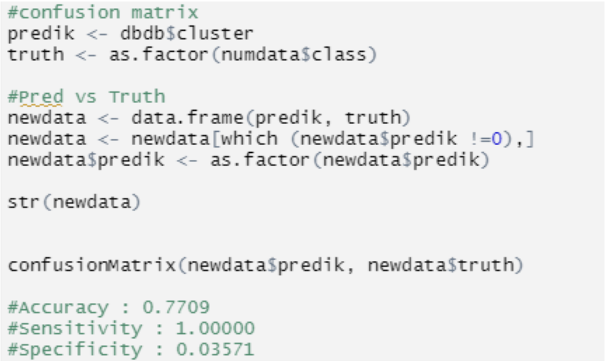
~~~
3) Visualisasi Data
```{r}
#menggunakan boxplot
boxplot(COD$timePlayed, main = "Distribusi Data Time Played", xlab
= "x", ylab = "y", col = c("cyan"), horizontal = TRUE, outline =
FALSE)

#menggunakan Histogram
library(dplyr)
library(tidyr)
numdata %>% gather(Attributes, value, 1:8) %>%
         ggplot(aes(x = value, fill = Attributes)) +
geom_histogram(colour = "black", show.legend = FALSE) +
facet_wrap(~Attributes, scales="free_x") + labs(x="Distribusi COD
PLAYER SKILLSS", y="Frekuensi", title="COD Attributes -
Histograms") + theme_bw()

#menggunakan scater plot
plot(numdata$timePlayed, numdata$wins, xlab = "timePlayed", ylab =
"wins", main = "Plot timeplayed vs wins", col = "sky blue")

# korelasi antarvariabel menggunakan scaterplot
ggpairs(cbind(numdata), lower=list(continuous="points"),
upper=list(continuous="blank"), axisLabels="none", switch="both") +
theme_bw()

#menggunakan density
dens <- density(numdata$timePlayed)
plot(dens,col="red")

#ggpairs
#install.packages('GGally')
library(GGally)
ggpairs(numdata)
```
~~~



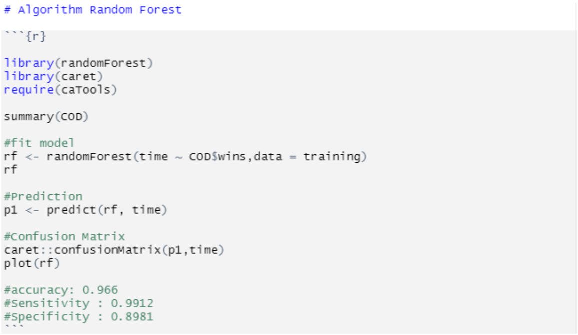
~~~
4) Eksplorasi Data
```{r}
library(ggplot2)
library(GGally)
library(tidyverse)
library(knitr)

#menampilkan struktur data
str(COD) #sebelum dipilih beberapa variabel
str(numdata) #variabel yg terpilih

#menampilkan 6 baris data pertama
head(numdata)

#menampilkan 6 baris data terakhir
tail(numdata)

#menampilkan summary data
summary(numdata)

```
~~~



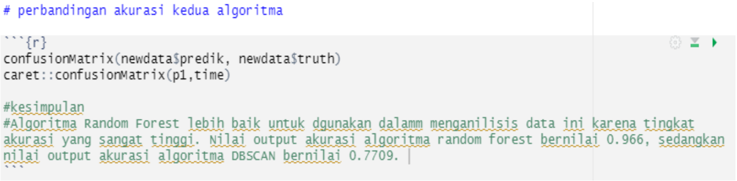
~~~
#Algorithm DBSCAN
```{r}
library(fpc)
library(factoextra)
library(tidyverse)
library(ggplot2)
library(gridExtra)
library(caret)


#calculate suitable epsilon
dbscan::kNNdistplot(numdata[1:8], k = 2)
epsilon <- 25000
abline(h = epsilon, lty = 2)
# eps is roughly at 25000


#cluster plot
dbb2 <- fpc::dbscan(numdata[1:8], eps = epsilon, MinPts = 5)
dbb2
dbdb <- dbscan::dbscan(numdata[1:8], eps = epsilon, minPts = 5)
dbdb
factoextra::fviz_cluster(dbdb, data = numdata[1:8], show.clust.cent = TRUE, geom = "point",
palette = "jco", ggtheme = theme_classic())



g <- ggplot(numdata, aes(level, wins )) +
  labs(col = "TIME PLAYED")

g1 <- g + geom_point(aes(col = numdata$timePlayed)) + ggtitle("Original Data")
g2 <- g + geom_point(aes(col = factor(dbb2$cluster+1))) + ggtitle("fpc")
g3 <- g + geom_point(aes(col = factor(dbdb$cluster+1))) + ggtitle("dbscan")

gridExtra::grid.arrange(g1,g2,g3, nrow = 3)
~~~



```
#confusion matrix
predik <- dbdb$cluster
truth <- as.factor(numdata$class)

#Pred vs Truth
newdata <- data.frame(predik, truth)
newdata <- newdata[which (newdata$predik !=0),]
newdata$predik <- as.factor(newdata$predik)

str(newdata)


confusionMatrix(newdata$predik, newdata$truth)

#Accuracy : 0.7709
#Sensitivity : 1.00000
#Specificity : 0.03571
```



~~~
# Algorithm Random Forest

```{r}

library(randomForest)
library(caret)
require(caTools)

summary(COD)

#fit model
rf <- randomForest(time ~ COD$wins,data = training)
rf

#Prediction
p1 <- predict(rf, time)

#Confusion Matrix
caret::confusionMatrix(p1,time)
plot(rf)

#accuracy: 0.966
#Sensitivity : 0.9912
#Specificity : 0.8981
```
~~~



~~~
# perbandingan akurasi kedua algoritma

```{r}
confusionMatrix(newdata$predik, newdata$truth)
caret::confusionMatrix(p1,time)

#kesimpulan
#Algoritma Random Forest lebih baik untuk dgunakan dalamm menganilisis data ini karena tingkat
akurasi yang sangat tinggi. Nilai output akurasi algoritma random forest bernilai 0.966, sedangkan
nilai output akurasi algoritma DBSCAN bernilai 0.7709.
```
~~~