# Project to perform steps in Data Mining Pipeline with eSports data

Varsha Sajja [2029023]

6/06/2021

## STEP a – Data Gathering and Integration

Two different datasets are selected from Kaggle - https://www.kaggle.com/jackdaoud/esports-earningsfor-players-teams-by-game related to eSports. The datasets illustrate the information on earnings of eSports players and teams. The data consists of earnings as per the game/genre.

Dataset 1 – highest_earning_players – Consists of players data mostly related to game.

Dataset 2 – highest_earning_teams – Consists of team details mostly related to game.

Two datasets can be merged (highest_earnings) to check the performances in each game relative to the player or the team.

```
>highest_earnings <- highest_earning_teams %>% inner_join(highest_earning_players, by="Game")
```

```
> head(highest_earnings)
  TeamId          TeamName TotalUSDPrize.x TotalTournaments     Game           Genre.x PlayerId    NameFirst NameLast CurrentHandle CountryCode TotalUSDPrize.y            Genre.y
1    760 San Francisco Shock         3105000                7 Overwatch First-Person Shooter    32000     Dong Jun      Kim        Rascal          kr        331108.7 First-Person Shooter
2    760 San Francisco Shock         3105000                7 Overwatch First-Person Shooter    40261      Nam Joo     Kwon       Striker          kr        327424.2 First-Person Shooter
3    760 San Francisco Shock         3105000                7 Overwatch First-Person Shooter    46828  Myeong Hwan      Yoo         smurf          kr        322184.2 First-Person Shooter
4    760 San Francisco Shock         3105000                7 Overwatch First-Person Shooter    36883      Hyo Bin     Choi    ChoiHyoBin          kr        319657.2 First-Person Shooter
5    760 San Francisco Shock         3105000                7 Overwatch First-Person Shooter    35563        Grant     Espe          Moth          us        314548.2 First-Person Shooter
6    760 San Francisco Shock         3105000                7 Overwatch First-Person Shooter    35121      Matthew   DeLisi         super          us        312948.0 First-Person Shooter
```

The above dataset represents the collective information of both teams and its players for each game type and genre.

## STEP b – Data Exploration

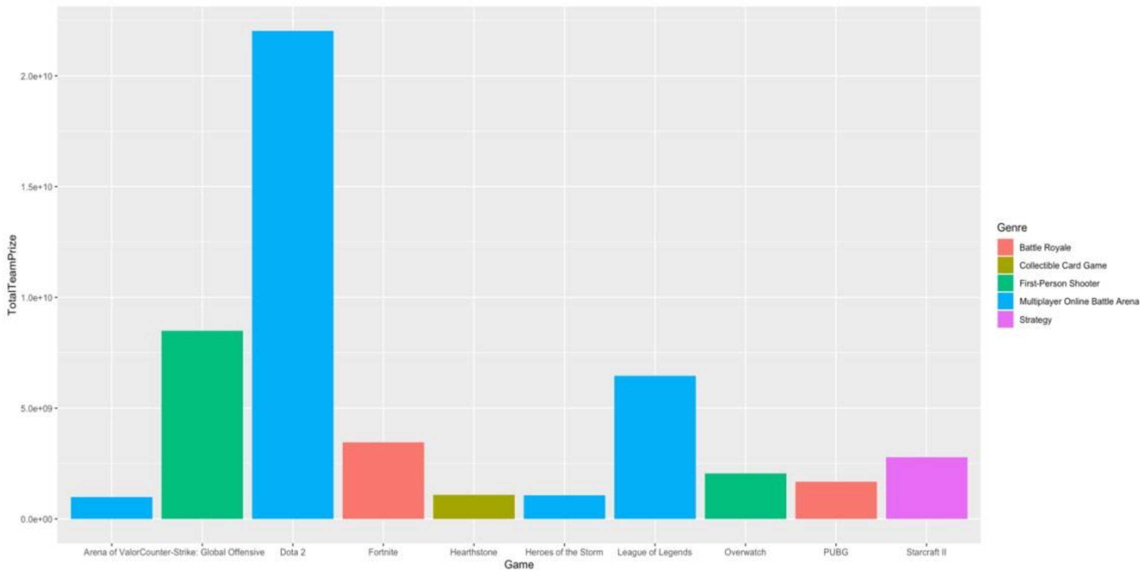
In this dataset, we have *92800* rows with *13* variables where *Genre* is redundant in merging. Now during this step, we summarize the statistical data and understand their distributions using Visualizations.

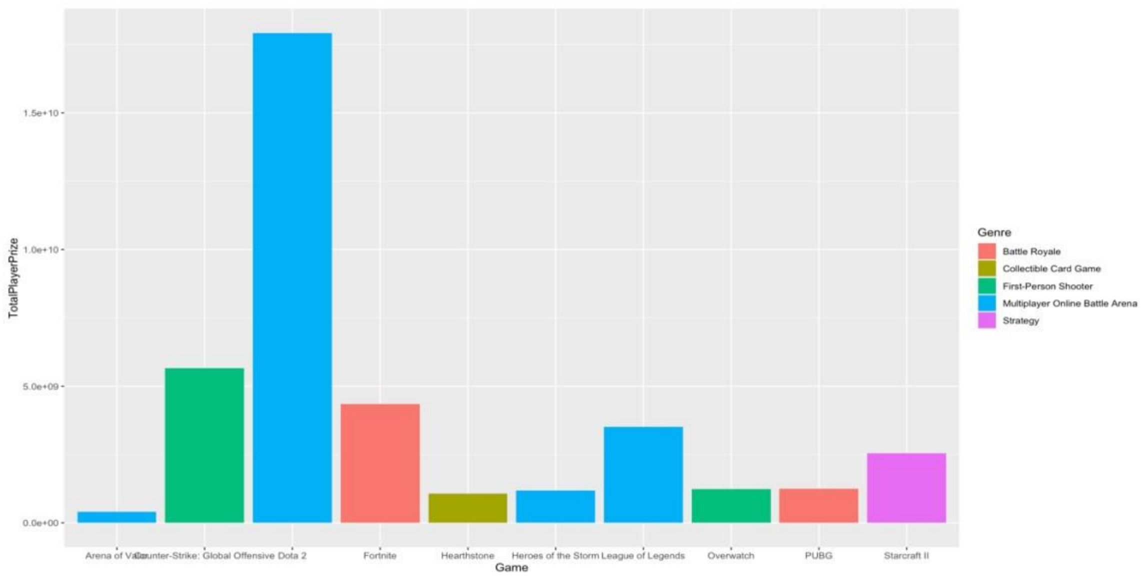
Let's investigate on different relationships between variables and their distributions.

1) Distribution of Game Vs. TotalTeamPrize and Game Vs. TotalPlayerPrize in relation with Genre.

   Here, most of the games played are in the genre "Multiplayer Online Battle Arena", also bags with highest earnings. Whereas "Collectible Card Game" genre is least played amongst others having least earnings.

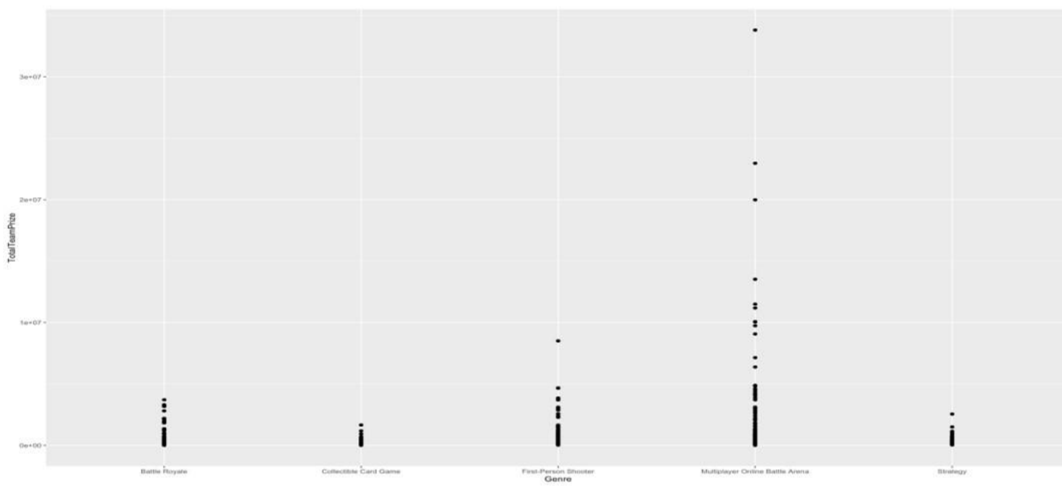
   Another observation is that the earnings are in alignment with players and teams accordingly.

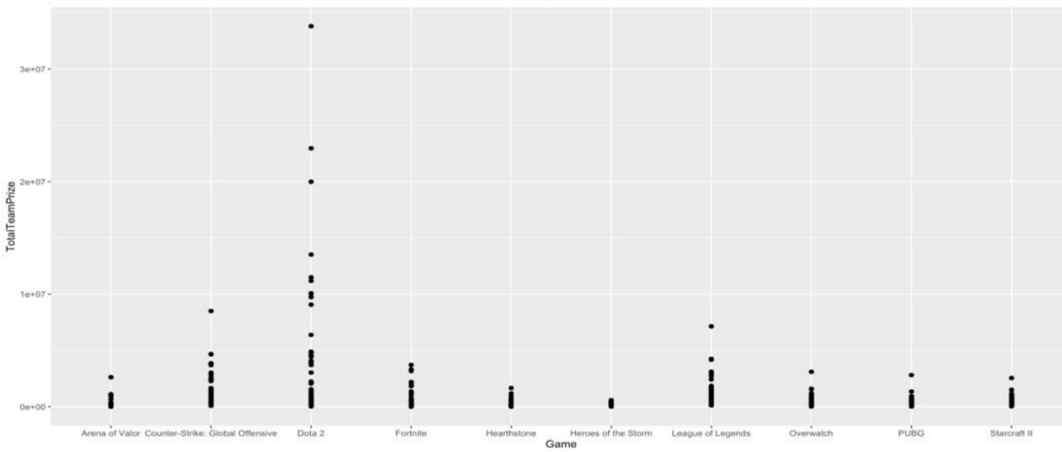


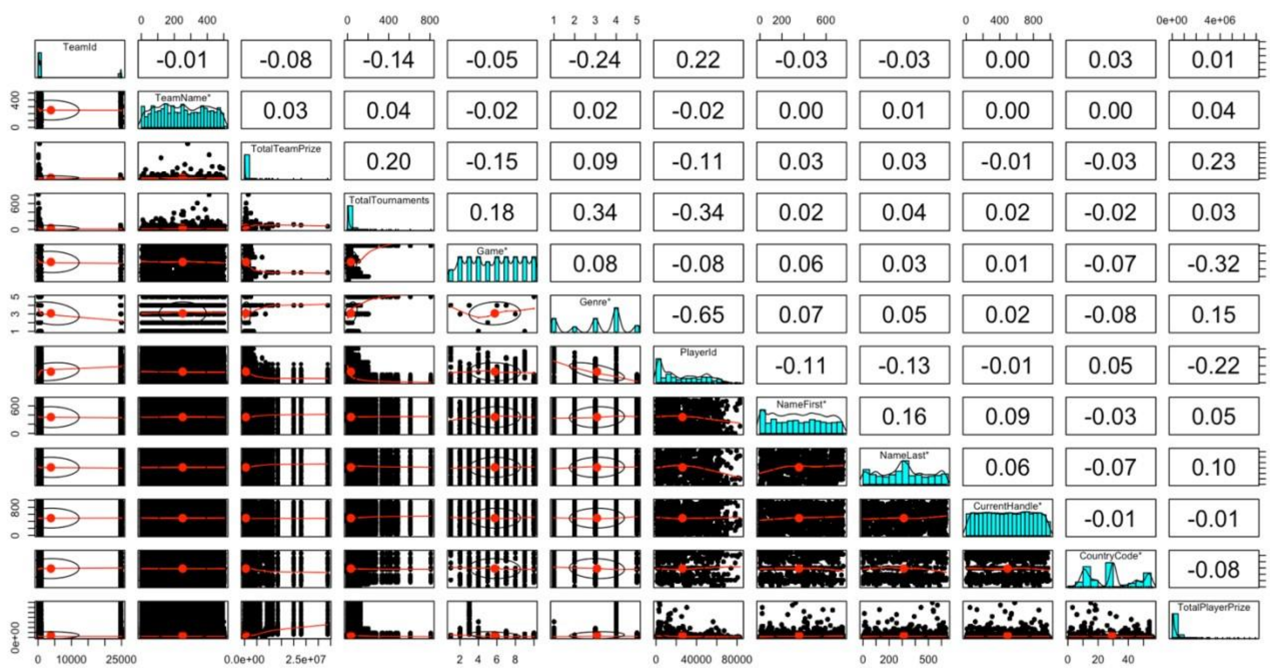

2) Distribution of Team earnings Vs. Game and Team earnings Vs. Genre

   Here we have 10 Game modes and 5 Genre types. From earlier distribution it is clear that earnings of team align with its players, thus the below plots gives information of earnings during games and genres separately. Team who played Dota2 had high earnings which is from highest earning Genre “Multiplayer Online Battle Arena”.




3) Distributions of all the variables

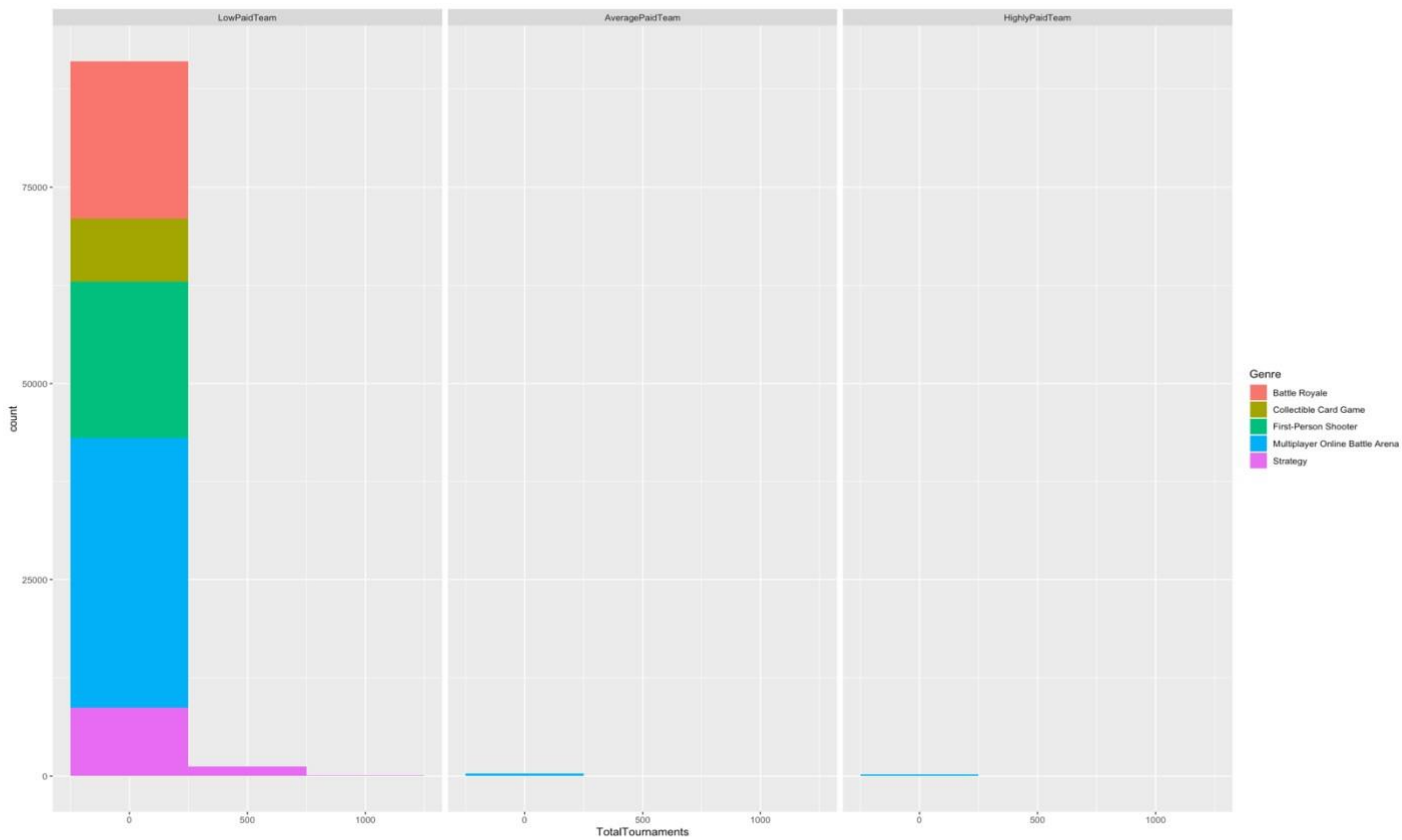


## STEP c – Data Cleaning

This is an essential step in the pipeline. Although, the selected dataset doesn't have redundant/missing values. It appears that the initial dataset is all cleaned one and merging is done to check the earnings of both teams and players. Further with the cleaned dataset, binning is performed on variables TotalUSDPrize.x and TotalUSDPrize.y. Kindly note that few of the variables are removed from dataset (earnings) as they are least significant with our target variable *Genre*.

```
>earnings <- highest_earnings %>%
  mutate(TeamPrizeStats = cut(TotalTeamPrize, breaks = 3,
                        labels=c("LowPaidTeam","AveragePaidTeam","HighlyPaidTeam")))
>earnings <- earnings %>%
  mutate(PlayerPrizeStats = cut(TotalPlayerPrize, breaks = 3,
                     labels=c("LowPaidPlayer","AveragePaidPlayer","HighlyPaidPlayer")))
```

On the cleaned data, visualization is performed to check who played more and which genre was highly paid.




The result from above plot also suggests as earlier conclusions that "Multiplayer Online Battle Arena" is played more number of times and paid more than all other genres.

## STEP d – Data Preprocessing

Basic steps in preprocessing the data resulted in Normalizing it as below.

```
> summary(norm1)
     TeamId           TeamName         TotalTournaments       Game              Genre              PlayerId         NameFirst        
 Min.   :-0.4430   Length:92800       Min.   :-0.50286   Length:92800       Length:92800       Min.   :-1.1781   Length:92800      
 1st Qu.:-0.4280   Class :character   1st Qu.:-0.45371   Class :character   Class :character   1st Qu.:-1.0043   Class :character  
 Median :-0.3922   Mode  :character   Median :-0.33904   Mode  :character   Mode  :character   Median :-0.2315   Mode  :character  
 Mean   : 0.0000                      Mean   : 0.00000                                         Mean   : 0.0000                     
 3rd Qu.:-0.3614                      3rd Qu.: 0.02136                                         3rd Qu.: 0.9007                     
 Max.   : 2.5089                      Max.   :12.71727                                         Max.   : 2.6643                     
   NameLast         CurrentHandle       CountryCode              TeamPrizeStats            PlayerPrizeStats 
 Length:92800       Length:92800       Length:92800       LowPaidTeam    :92300   LowPaidPlayer    :90700  
 Class :character   Class :character   Class :character   AveragePaidTeam:  300   AveragePaidPlayer: 1400  
 Mode  :character   Mode  :character   Mode  :character   HighlyPaidTeam :  200   HighlyPaidPlayer :  700  
```

As we have much of categorical variables in our data, I went ahead by taking dummies.

```
>dummy <- dummyVars(Genre ~ ., data = earnings1)
>dummies <- as.data.frame(predict(dummy, newdata = earnings1))
>head(dummies)
```

The dummies are resulted as below which can be used for Principal Component Analysis later.

```
> head(dummies)
  TotalTournaments GameArena of Valor GameCounter-Strike: Global Offensive GameDota 2 GameFortnite GameHearthstone GameHeroes of the Storm GameLeague of Legends GameOverwatch GamePUBG GameStarcraft II PlayerId
1                7                  0                                    0          0            0               0                       0                     0             1        0                0    32000
2                7                  0                                    0          0            0               0                       0                     0             1        0                0    40261
3                7                  0                                    0          0            0               0                       0                     0             1        0                0    46828
4                7                  0                                    0          0            0               0                       0                     0             1        0                0    36883
5                7                  0                                    0          0            0               0                       0                     0             1        0                0    35563
6                7                  0                                    0          0            0               0                       0                     0             1        0                0    35121
  TeamPrizeStats.LowPaidTeam TeamPrizeStats.AveragePaidTeam TeamPrizeStats.HighlyPaidTeam PlayerPrizeStats.LowPaidPlayer PlayerPrizeStats.AveragePaidPlayer PlayerPrizeStats.HighlyPaidPlayer
1                          1                              0                             0                              1                                  0                                 0
2                          1                              0                             0                              1                                  0                                 0
3                          1                              0                             0                              1                                  0                                 0
4                          1                              0                             0                              1                                  0                                 0
5                          1                              0                             0                              1                                  0                                 0
6                          1                              0                             0                              1                                  0                                 0
> 
```

## STEP e – Clustering

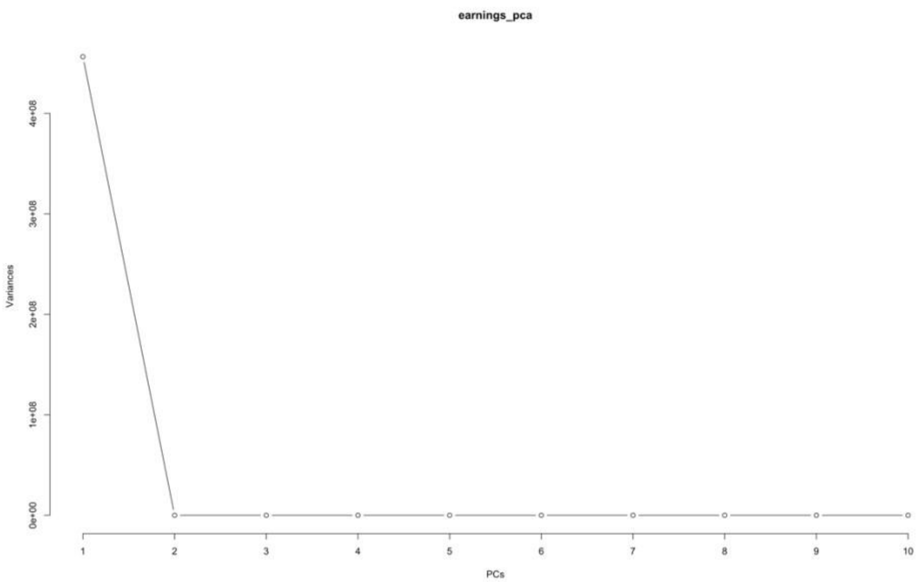
For this step, as the dataset is huge, I have analyzed the process by taking the test set.

```
>train <- earnings %>% sample_frac(.75)    >test  <-
anti_join(earnings, train)
```

PCA is used on the data (dummies) and the result is reported as below.

```
> summary(earnings_pca)
Importance of components:
                         PC1      PC2    PC3    PC4    PC5    PC6    PC7    PC8    PC9   PC10   PC11   PC12    PC13   PC14    PC15
Standard deviation     21363 57.38334 0.3388 0.3283 0.3283 0.3283 0.3283 0.3036 0.2865 0.2521 0.1652 0.1493 0.09735 0.0879 0.05036
Proportion of Variance     1  0.00001 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.00000 0.0000 0.00000
Cumulative Proportion      1  1.00000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.00000 1.0000 1.00000
                            PC16      PC17      PC18
Standard deviation     1.108e-13 4.894e-15 2.454e-15
Proportion of Variance 0.000e+00 0.000e+00 0.000e+00
Cumulative Proportion  1.000e+00 1.000e+00 1.000e+00
```

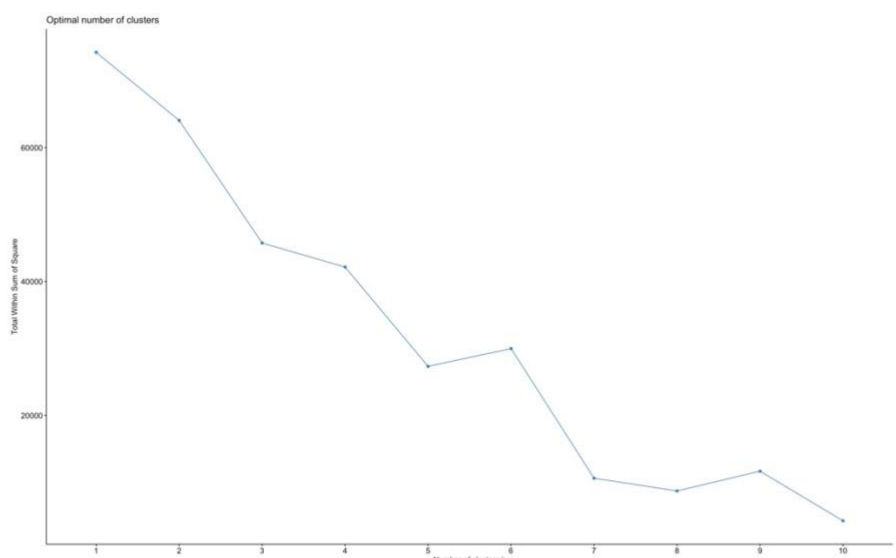
By looking into the proportion, we can say that 100% of variance is captured in first principal component itself. Elbow graph plotted below also suggests the same.



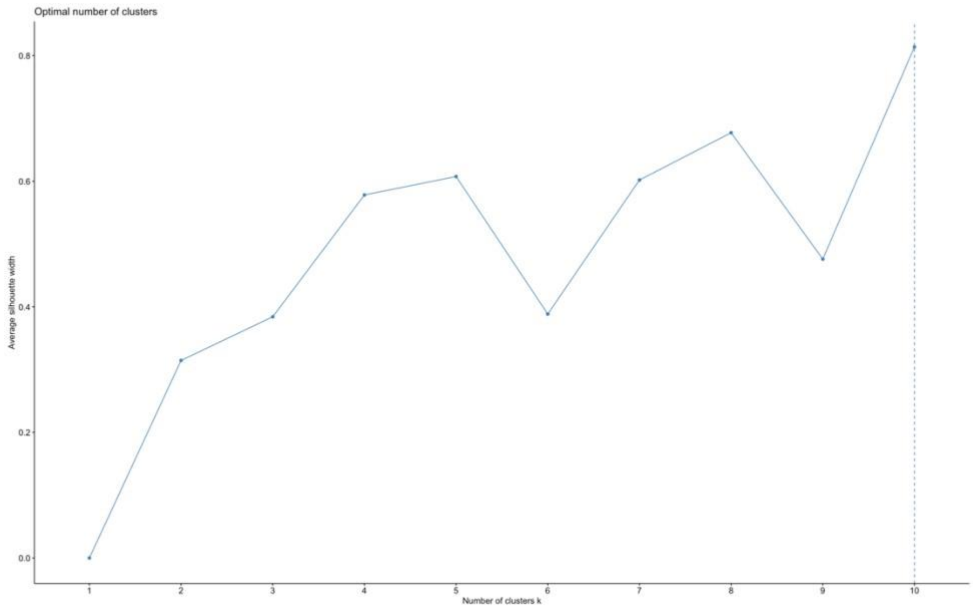

By using SVM method, we get to check accuracy with respect to Genre. The accuracy is reported to be 100% as below.

```
> svm1
Support Vector Machines with Linear Kernel

92800 samples
   18 predictor
    5 classes: 'Battle Royale', 'Collectible Card Game', 'First-Person Shooter', 'Multiplayer Online Battle Arena', 'Strategy'

No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 74240, 74240, 74240, 74240, 74240
Resampling results:

  Accuracy  Kappa
  1         1

Tuning parameter 'C' was held constant at a value of 1
```

Now let try k-means clustering technique on the data.

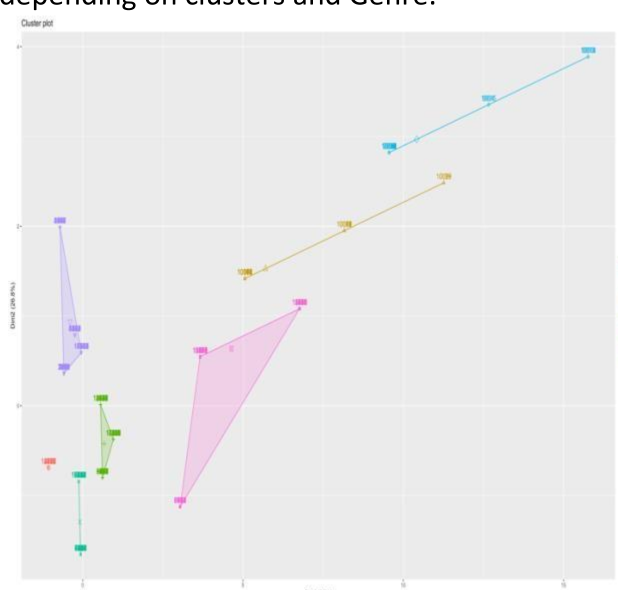


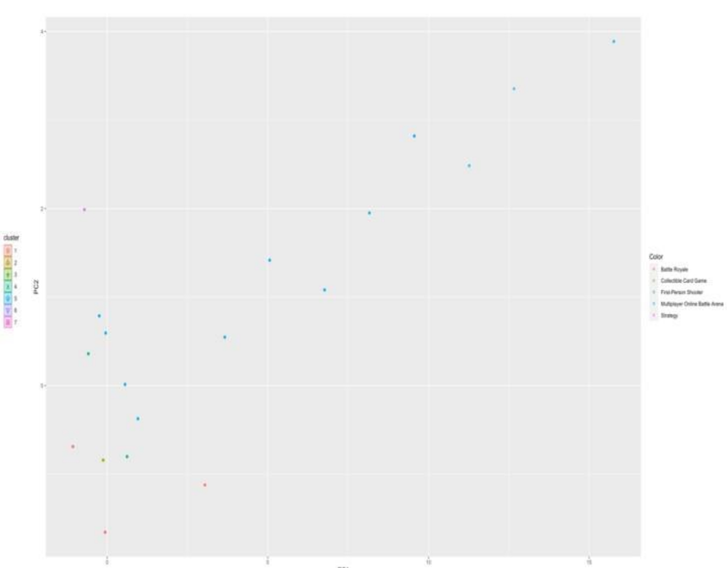
The above plot is dynamic at every k value. But you can see the linear trend after k=7. Thus, our k value is selected to be 7. Let's now take a chance to explore Silhouette method. The below plot suggest 10 clusters as optimal. We can now conclude that k=7 would be best fit.




```
>fit <- kmeans(predictors, centers = 7, nstart = 50)
```

Clustering has been performed using 7 clusters as optimal solution for this data. The result is as below depending on clusters and Genre.



**STEP f – Classification**

For predicting label *Genre* in my data, I used two classifiers k nearest neighbor and decision tree where kNN stood as good fit because decision tree prediction accuracy is worst as compared to kNN.

kNN classification

```
>knnFit <- train(Genre ~ ., data = train1,
method = "knn",                trControl
= ctrl,
                preProcess = c("center","scale"))
```

```
> knnFit
k-Nearest Neighbors

83520 samples
    4 predictor
    5 classes: 'Battle Royale', 'Collectible Card Game', 'First-Person Shooter', 'Multiplayer Online Battle Arena', 'Strategy'

Pre-processing: centered (12), scaled (12)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 75167, 75169, 75168, 75169, 75168, 75168, ...
Resampling results across tuning parameters:

  k  Accuracy  Kappa
  5  1         1
  7  1         1
  9  1         1

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

Accuracy for kNN fit is reported to be 100% for optimal value of k as 9. We can further check this using confusion matrix as below.

```
> confusionMatrix(test1$Genre, pred_earn)
Confusion Matrix and Statistics

                                Reference
Prediction                       Battle Royale Collectible Card Game First-Person Shooter Multiplayer Online Battle Arena Strategy
  Battle Royale                           1782                     0                    0                              0        0
  Collectible Card Game                      0                   627                    0                              0        0
  First-Person Shooter                       0                     0                 1800                              0        0
  Multiplayer Online Battle Arena            0                     0                    0                           3206        0
  Strategy                                   0                     0                    0                              0      983

Overall Statistics

               Accuracy : 1
                 95% CI : (0.9996, 1)
    No Information Rate : 0.3818
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
```

Decision Tree

```
>dtree <- train(Genre ~., data = train1, method = "rpart", trControl = ctrl)
```

```
> dtree
CART

83520 samples
    4 predictor
    5 classes: 'Battle Royale', 'Collectible Card Game', 'First-Person Shooter', 'Multiplayer Online Battle Arena', 'Strategy'

No pre-processing
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 75168, 75169, 75167, 75168, 75167, 75168, ...
Resampling results across tuning parameters:

  cp         Accuracy   Kappa
  0.0000000  1.0000000  1.00000000
  0.1552239  1.0000000  1.00000000
  0.1724454  0.4062992  0.06258059

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.1552239.
```

The accuracy for decision tree classifier is 100% at cp=0.15 but confusion matrix gives us only 50% accuracy which is not a good fit for the label *Genre.*

```
> confusionMatrix(test1$Genre, pred_earn2)
Confusion Matrix and Statistics

                                Reference
Prediction                       Battle Royale Collectible Card Game First-Person Shooter Multiplayer Online Battle Arena Strategy
  Battle Royale                              0                     0                    0                            1782        0
  Collectible Card Game                      0                     0                    0                             627        0
  First-Person Shooter                       0                     0                    0                            1800        0
  Multiplayer Online Battle Arena            0                     0                    0                            3206        0
  Strategy                                   0                     0                    0                               0      983

Overall Statistics

               Accuracy : 0.4988
                 95% CI : (0.4881, 0.5096)
    No Information Rate : 0.8829
    P-Value [Acc > NIR] : 1

                  Kappa : 0.228
```

## STEP g – Evaluation

1) From the earlier step, I have concluded that kNN is best fit as a classifier to the data. Since the *Genre* has more than 2 classes, I have transformed it to two classes classifying into Online and Offline genres.

```
>test_model <- test1
>offline <- c("First-Person Shooter", "Collectible Card Game", "Strategy")
>online <- c("Multiplayer Online Battle Arena", "Battle Royale")
>GenreMode <- rbin_factor_combine(test_model, Genre, offline, "OFFLINE")
>GenreMode <- rbin_factor_combine(GenreMode, Genre, online, "ONLINE")
```

Now our new transformed dataset is *GenreMode* which consists of 2 classes in *Genre.* Again applying kNN on *GenreMode* gives us the same result as 100% accuracy for k=9.

```
> knnFit2
k-Nearest Neighbors

8391 samples
   4 predictor
   2 classes: 'ONLINE', 'OFFLINE'

Pre-processing: centered (12), scaled (12)
Resampling: Cross-Validated (10 fold)
Summary of sample sizes: 7551, 7552, 7552, 7552, 7553, 7552, ...
Resampling results across tuning parameters:

  k  Accuracy  Kappa
  5  1         1
  7  1         1
  9  1         1

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was k = 9.
```

2*2 confusion matrix for *Genre* using data *GenreMode* is as below.

```
> confusionMatrix(GenreMode$Genre, pred_earn3)
Confusion Matrix and Statistics

          Reference
Prediction ONLINE OFFLINE
   ONLINE    4967       0
   OFFLINE      0    3424

               Accuracy : 1
                 95% CI : (0.9996, 1)
    No Information Rate : 0.5919
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 1
```

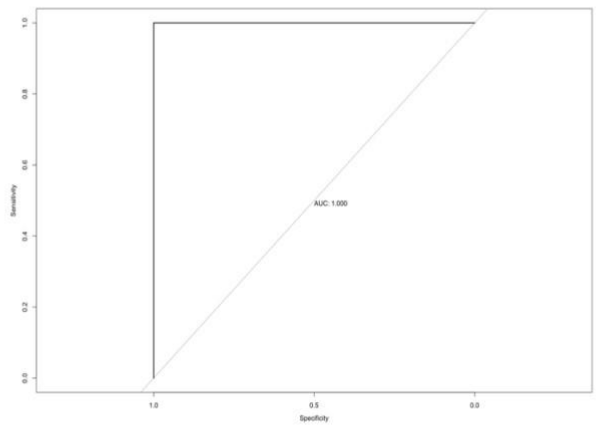
2) Precision and recall are calculated for individual classes as below.

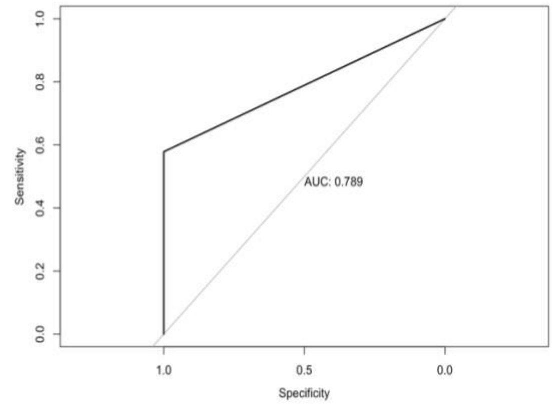
```
> # Get the precision value for each class
> metrics1 %>% select(c(Precision))
                                      Precision
Class: Battle Royale                          1
Class: Collectible Card Game                  1
Class: First-Person Shooter                   1
Class: Multiplayer Online Battle Arena        1
Class: Strategy                               1
> # Get the recall value for each class
> metrics1 %>% select(c(Recall))
                                      Recall
Class: Battle Royale                       1
Class: Collectible Card Game               1
Class: First-Person Shooter                1
Class: Multiplayer Online Battle Arena     1
Class: Strategy                            1
```

3) ROC plot

kNN – best model

decision tree




From the 2*2 confusion matrix it is clear that the ONLINE class is having higher count than OFFLINE where 3 genres are combined. It indicates that ONLINE class – Multiplayer Online Battle Arena, Battle Royale is most played than others. In the initial analysis of visualizations, we have concluded that the highest earnings are mostly from Multiplayer online Battle Arena which supports the comparison here. The accuracy also fits exactly.

**STEP h – Report**

Takeaways from the data and Analysis includes,

- The raw data consists of 2 datasets: highest_earning_players, highest_earning_teams having common variables as Genre and Game. Thus, both datasets have been merged to find which Genre has most players and highest earnings.
- Analysis has been initially performed on merged dataset: highest_earnings (92800 rows, 12 variables) which has been later transformed as per required analysis.
- The dataset is cleaned and further no actions have been performed towards cleaning it. Instead data binning has been done due to significant analysis to be performed on *Genre.* Also, data type for few significant variables (Game, Genre) have been transformed into factors. Normalization process is applied on the data and dummies are created to check for PCA.
- Principal Component Analysis resulted in 100% of variance around the data with 1 component itself. Scree plot also suggested the same.
- Kmeans clustering is used for this data and optimal clusters were reported as 7 after analysis.
- During different classifications, kNN worked best with the data where partitioning of data is done beforehand. Accuracy was reported to be 100% for k=9.
- During different classifications, decision tree gave the similar accuracy but its performance is bad as we predict from confusion matrix giving accuracy as only 50%.

- Evaluation of the model resulted in transforming the data into 2 classes by binning method where ONLINE class consists of Multiplayer Online Battle Arena and Battle Royale, OFFLINE class consists of First-Person Shooter, Collectible Card game and Strategy. The results are interesting and are similar to the initial analysis as the highest earnings, most played were from Online class.
- The conclusion from the analysis performed is that the highest played and earned genre is Multiplayer Online Battle Arena where the results also show same.

**STEP i – Reflection**

My learnings include from data cleaning to applications in data science. In the process, I have understood how to choose among different sets of data to work with (whether cleaned, semi-cleaned, dirty). Also, preprocessing methods give exact results which can be later used for clustering/ classification depending upon what we need from data and what data gives us. Interpreting those results makes easy for me to understand what is happening with the data completely. In the final week, my takeaways are beyond expectation as the ethics required during data mining is specifically concerned for any data analyst/scientist. By learning these concepts each week strengthened my decisions to work with algorithms for different types of data.