UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Faculty of Engineering,
Built Environment and
Information Technology

Fakulteit Ingenieurswese, Bou-omgewing en
Inligtingtegnologie / Lefapha la Boetšenere,
Tikologo ya Kago le Theknolotši ya Tshedimošo

Department of Informatics

INDIVIDUAL ASSIGNMENT

Surname	Wood							
Initials	FS							
Student Number	0	4	8	6	9	4	6	1
Module Code	**INF**				7	9	1	
Assignment number	Assignment 1							
Name of Lecturer	Dr Mike Nkongolo							
Date of Submission	2024/09/12							
Declaration: I declare that this assignment, submitted by me, is my own work and that I have referenced all the sources that I have used.								
Signature of Student	Fabio Wood							

Contents

600
500
400
300
200
100
0
JAN
FEB
MAR
APR
MAY
JUN
JUL
AUG
SEP
OCT
NOV
DEC

Title Page

Title: How can cybergaming analytics be assessed through learning in a gamified way

Student: Fabio Wood

Student Number: u04869461

Assignment Number: 1

Lecturer: Dr. Mike Nkongolo

Date of Submission: 17th September 2024

Abstract

This report analyses how the "CyberVigilance" game can offer cybersecurity education in a gamified learning environment. The main aim of this report is to display how the players behaved, along with the game balance and players cybersecurity awareness. This is done by using exploratory data and machine learning techniques. The data that was collected is from various players in the INF 791 class at the University of Pretoria. This data was then processed and analysed using various machine learning models. The results display that defenders have a very small advantage over the attackers. Nonetheless, the gameplay across all different levels of the game is balanced, except for the intermediate level. This is due to the prediction accuracy that the models struggled to obtain due to the lack of information in the intermediate level. The model best suited for this study was the ensemble model with a 99.6% accuracy. These findings are able to offer insights based on the "CyberVigilance" game, and in turn can allow for the generation of strategies for the improvement of cybersecurity awareness and engagement through gamification.

Introduction

Context:

To increase cybersecurity awareness, it is important for best practices, as well as education on cyber security risks to be taught. By using environments in which a user can play a game and learn about a specific topic, in this case cyber security, they are able to learn through practical skills and actively engage with cyber security concepts. This report explains how the "CyberVigilance" game has been designed to allow defense and attack scenarios for cyber security.

Problem Statement:

This report aims to look at player performance, gameplay and the effectiveness of learning cyber security concepts and strategies through game scenarios. In detail, it analyzes how well players can perform in both attacker and defender scenarios, and how machine learning is used in order to predict the outcomes of games and player behavior.

Objectives:

The objectives would be to evaluate how machine learning models are able to predict the outcomes of games. The ability to analyze the performance of players across multiple rounds and levels using the game. To evaluate the balance between attacker and defender roles. Finally, to provide insights into game improvements as well as player insights.

Scope:

The data set is made-up of performance metrics according to the players, their score, and the game durations. This report applies various techniques such as EDA and machine learning. The report can contribute to the understanding of cyber security awareness and education, as well as the proposal of improvements based on the various findings.

Literature Review

By playing games in relation to cybersecurity or computer security concepts , players are able to increase their awareness of security risks, as well as broaden their creative thinking (Denning et al., 2013). By combining theoretical lessons and gameplay, users are able to improve their understanding and retention (Yasin et al., 2018). In today’s world, more and more young adults, also known as *‘Tweens’* are facing cyber security threats and risks online (Maqsood & Chiasson, 2021). Luckily by applying gamification – a way of how game mechanics can engage users to learn through enjoyable ways – to security concepts, they are able to learn about these risks and how to mitigate them (Scholefield & Shepherd, 2019).

Data Collection and Preparation

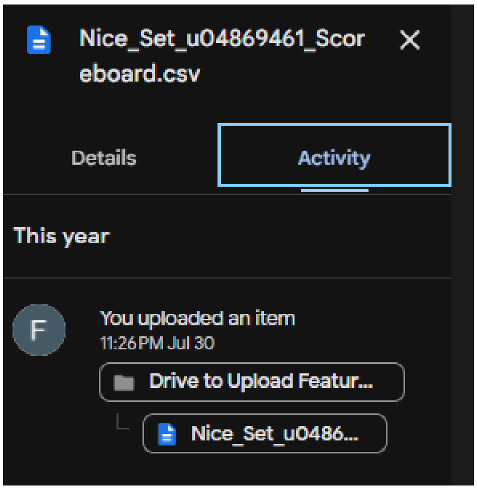
Phase 1 Data Sources

The required data fields, namely the Nickname, Defender Score, Attacker Score, Time (sec), Winner and Level were captured. These captured data fields all contained values for each column and cell, thereby completing the scoreboard. This scoreboard of mine was then exported in CSV format as done so by an ‘Export to CSV’ button within the game of “CyberVigilance”.

According to Assignment 1, the data that was collected is data that was submitted by each student in the class. However, this data came in all different types of formats and file extensions, as well as some of the data being incomplete in some respects with other students. It is important that this data is then collected, cleaned, analysed and processed.

Phase 2: Data Description

This phase mainly consisted of the upload of my CSV file into the designated drive before the assigned deadline which was the 2nd of August 2024. This CSV file was uploaded to the designated drive on the 30th of July 2024 as displayed below:



Nice_Set_u04869461_Scoreboard.csv

This year

You uploaded an item
11:26 PM Jul 30

Within the “CyberVigilance” game set out by Dr Nkongolo, the students each had to download the game, as well as play it in Visual Studio Code. This then resulted in each student receiving a scoreboard which populated columns namely Nickname, Defender Score, Attacker Score, Time (sec), Winner and Level. This scoreboard was then able to be exported to a CSV format and uploaded into the designated *Google Drive*. From this *Google Drive* students then had to download all the datasets that were uploaded by each student. I created a script that was able to go through each one of the datasets and retrieve the data into one file.

Phase 3: Data Preprocessing and EDA

The datasets were able to be downloaded, and I created a script that was able to combine all the datasets into one. The script then went on to make sure to drop all duplicates, remove any negative values from the time feature and drop rows where the Nickname was missing. Due to there not being many missing values in the Defender Score, Attacker Score or Time, I set the script to replace any of these missing cells with a 0. The script went onto save the cleaned data into a new CSV which I called ‘Cleaned_Game_Analytics_Dataset.csv’. From this new dataset I was then able to continue with the data processing.

Methodology

Tools and Libraries

Pip Installs:

- pip install pandas
- pip install numpy
- pip install matplotlib
- pip install seaborn
- pip install scikit-learn
- pip install lazypredict

Packages:

- pandas
- numpy
- matplotlib
- seaborn
- scikit-learn
- lazypredict

Software:

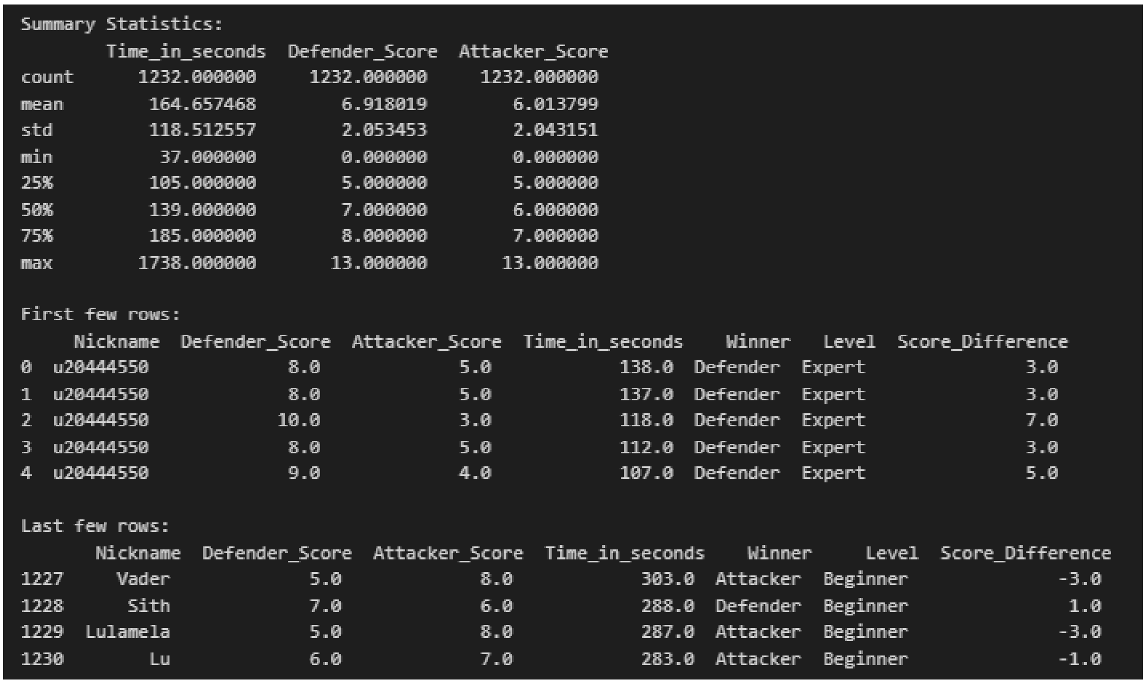
- Python 3.x
- IDE or Code Editor
- Notebook
- PyCharm
- Visual Studio Code

To visualize the data that was collected, processed, cleaned and then used, it is important to display this data visually to portray the results. These results are crucial to understand the data and are easier to interpret when displayed visually. I have broken it down further to understand the analysis more in depth:

Results

For the Dataset

Based on the dataset, I created a script that was able to generate the basic dataset statistics on the 'Cleaned_Game_Analytics_Dataset.csv'.



```
Summary Statistics:
       Time_in_seconds  Defender_Score  Attacker_Score
count      1232.000000     1232.000000     1232.000000
mean        164.657468        6.918019        6.013799
std         118.512557        2.053453        2.043151
min          37.000000        0.000000        0.000000
25%         105.000000        5.000000        5.000000
50%         139.000000        7.000000        6.000000
75%         185.000000        8.000000        7.000000
max        1738.000000       13.000000       13.000000

First few rows:
    Nickname  Defender_Score  Attacker_Score  Time_in_seconds    Winner   Level  Score_Difference
0  u20444550             8.0             5.0            138.0  Defender  Expert               3.0
1  u20444550             8.0             5.0            137.0  Defender  Expert               3.0
2  u20444550            10.0             3.0            118.0  Defender  Expert               7.0
3  u20444550             8.0             5.0            112.0  Defender  Expert               3.0
4  u20444550             9.0             4.0            107.0  Defender  Expert               5.0

Last few rows:
       Nickname  Defender_Score  Attacker_Score  Time_in_seconds    Winner     Level  Score_Difference
1227      Vader             5.0             8.0            303.0  Attacker  Beginner              -3.0
1228       Sith             7.0             6.0            288.0  Defender  Beginner               1.0
1229   Lulamela             5.0             8.0            287.0  Attacker  Beginner              -3.0
1230         Lu             6.0             7.0            283.0  Attacker  Beginner              -1.0
```

The key concepts to note for each of these numerical columns are:

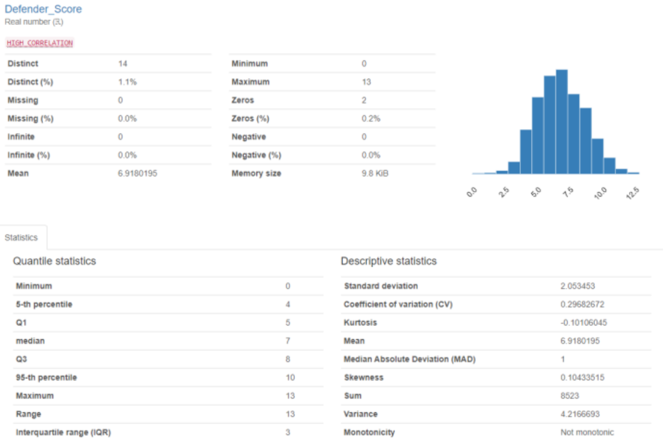
- **Time_in_seconds:**
    - Mean: 164.66 seconds
    - Standard Deviation: 118.51 seconds
    - Min: 37 seconds
    - Max: 1738 seconds
- **Defender_Score:**
    - Mean: 6.91
    - Standard Deviation: 2.05
    - Min: 0
    - Max: 13
- **Attacker_Score:**
    - Mean: 6.01
    - Standard Deviation: 2.04
    - Min: 0
    - Max: 13

The key insights that can be acquired from this data is the wide range in terms of the time it took a student or player to complete a game. This may be an indication of a player strategy, or a learning curve for the student throughout the game. The defender score and attacker scores are close according to the average, however, the minimum of a 0 score for both the attacker and defender could indicate that the student was not engaged or not learning throughout the game. Lastly moving onto the standard deviation. According to the standard deviation for both the attacker and defender scores, we notice there is an even spread in terms of how each of the students can perform the role of the attacker or the defender. The attacker and defender roles having a maximum of 13 each (highest you could get), is evident that the student learnt throughout the game, but indicated that there is no advantage to being an attacker or a defender in terms of the game involving cybersecurity.

In addition to these statistics, I have created an EDA report that can be viewed with the following link:

file:///C:/Users/fabio/OneDrive/Desktop/Fabio%20Honors%202024/SEMESTER%202/INF%20791/ASSIGNMENT%201/EDA_Report.html

Screenshots of this report pertaining to the above information are included below:



Defender_Score
Real number (ℝ)

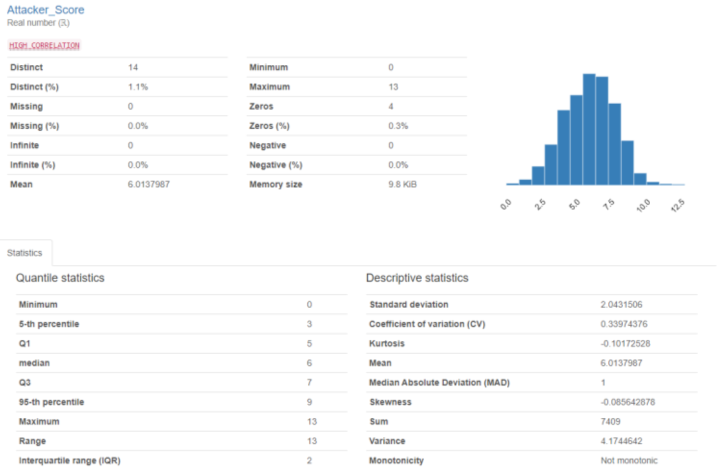
HIGH CORRELATION

Distinct	14	Minimum	0
Distinct (%)	1.1%	Maximum	13
Missing	0	Zeros	2
Missing (%)	0.0%	Zeros (%)	0.2%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	6.9180195	Memory size	9.8 KiB

Statistics

Quantile statistics		Descriptive statistics	
Minimum	0	Standard deviation	2.053453
5-th percentile	4	Coefficient of variation (CV)	0.29682672
Q1	5	Kurtosis	-0.10106045
median	7	Mean	6.9180195
Q3	8	Median Absolute Deviation (MAD)	1
95-th percentile	10	Skewness	0.10433515
Maximum	13	Sum	8523
Range	13	Variance	4.2166693
Interquartile range (IQR)	3	Monotonicity	Not monotonic

Something tells me you should take cyber security more seriously.



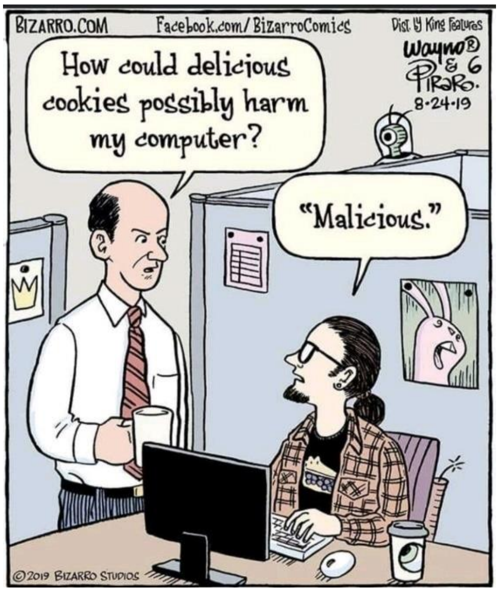
Attacker_Score
Real number (ℝ)

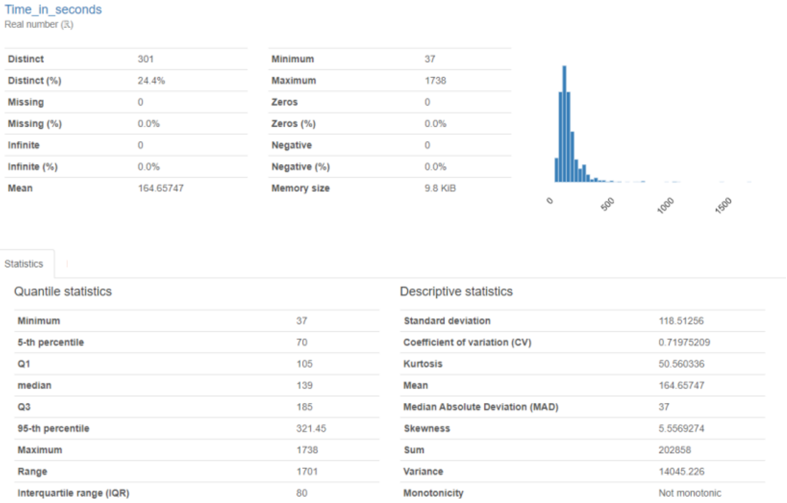
HIGH CORRELATION

Distinct	14	Minimum	0
Distinct (%)	1.1%	Maximum	13
Missing	0	Zeros	4
Missing (%)	0.0%	Zeros (%)	0.3%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	6.0137987	Memory size	9.8 KiB

Statistics

Quantile statistics		Descriptive statistics	
Minimum	0	Standard deviation	2.0431506
5-th percentile	3	Coefficient of variation (CV)	0.33974376
Q1	5	Kurtosis	-0.10172528
median	6	Mean	6.0137987
Q3	7	Median Absolute Deviation (MAD)	1
95-th percentile	9	Skewness	-0.085642878
Maximum	13	Sum	7409
Range	13	Variance	4.1744642
Interquartile range (IQR)	2	Monotonicity	Not monotonic





Time_in_seconds
Real number (ℝ)

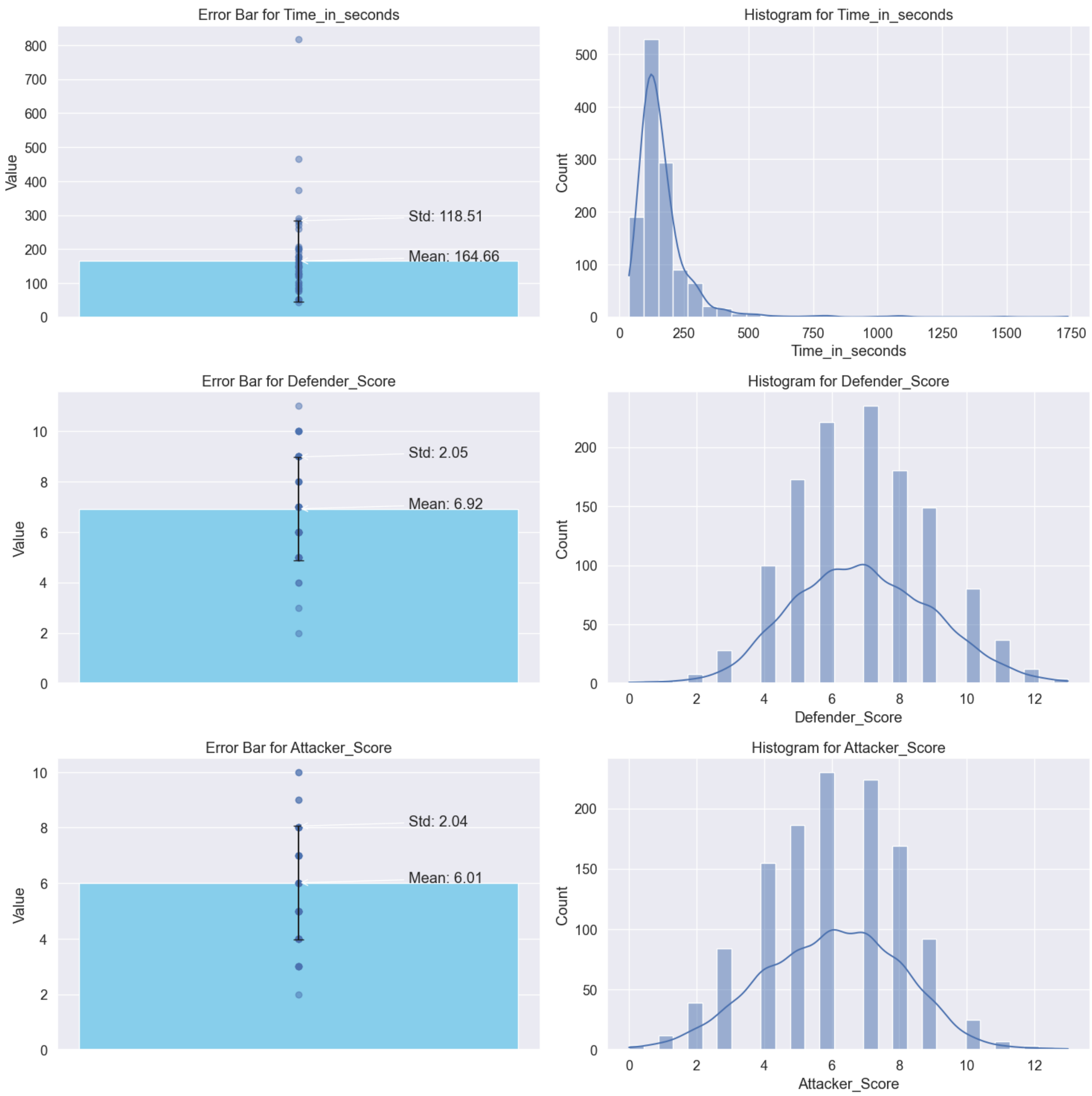
Distinct	301	Minimum	37
Distinct (%)	24.4%	Maximum	1738
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	0
Infinite (%)	0.0%	Negative (%)	0.0%
Mean	164.65747	Memory size	9.8 KiB

Statistics

Quantile statistics		Descriptive statistics	
Minimum	37	Standard deviation	118.51256
5-th percentile	70	Coefficient of variation (CV)	0.71975209
Q1	105	Kurtosis	50.560336
median	139	Mean	164.65747
Q3	185	Median Absolute Deviation (MAD)	37
95-th percentile	321.45	Skewness	5.5569274
Maximum	1738	Sum	202858
Range	1701	Variance	14045.226
Interquartile range (IQR)	80	Monotonicity	Not monotonic

For Numeric Values

It is important to note the numerical variables in this dataset. These include Defender Score, Attacker Score, Time (sec) and the Score Difference – which I created in addition to the previous columns.



Time_in_seconds:

- Error Bar: Due to the large standard deviation, this could indicate that the game times only vary slightly. The error bar can display all the points, and most of the points are within a reasonable range. The outliers indicate that there are some games that last a bit longer than others.
- Histogram: The histogram can show a that the data is skewed to the right, meaning that some most of the games are shorter. The long tail that is indicated in that diagram is due to some of the games that last for much longer.
- In conclusion, most of the games are very short, however, there is many games that finish quickly, with some outliers indicating that some students took longer to play.

Defender_Score:

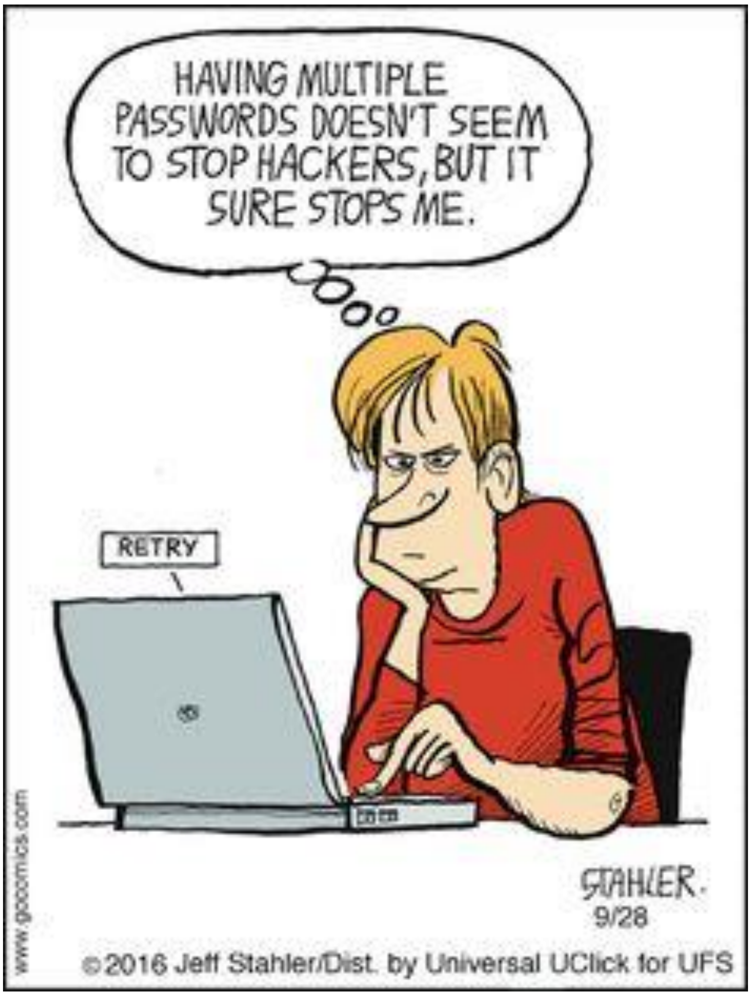
- Error Bar: The error bar suggests that most of the defenders scored between four to nine points.
- Histogram: The distribution seems to be normally distributed, which in turn means that most of the defenders were able to score between six and eight points.

- In conclusion, there is a consistency between the defender performance. This is turn can deduce that the game for the defenders is fair.

Attacker_Score:

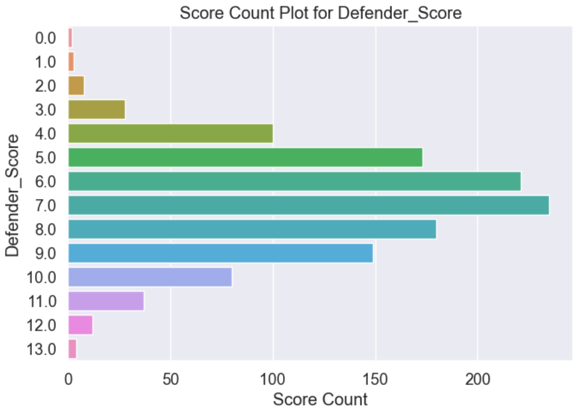
- Error Bar: Based on the standard deviation and the mean of the attacker, it is evident that the score is a bit lower than the defenders average, and the standard deviation is similar. This means that there is a very similar variation in the score of the two.
- Histogram: This score for the attacker seems to be normally distributed, just like that of the defender.
- In conclusion, it is evident that although the attacker performance is consistent, according to the average, the defenders might have a very small advantage in the ability to obtain scores.

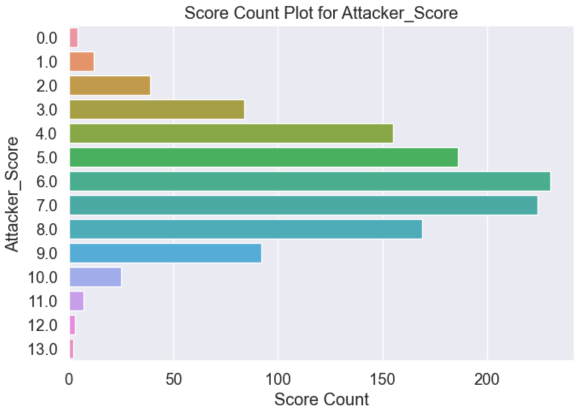
Based on the above graphs and insights, it is noted that although most of the games are fast, there are some games that last longer than others. Regardless of this, the attackers and defenders seem to have similar scoring capabilities, however, the defenders have a slight upper hand. As both the attacker score and defender score are normally distributed, the game for both is balanced.

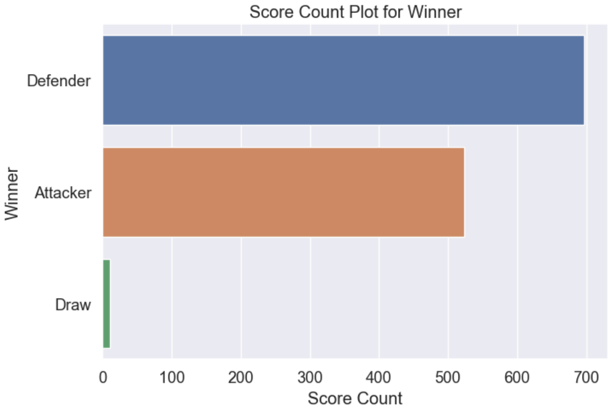

HAVING MULTIPLE PASSWORDS DOESN'T SEEM TO STOP HACKERS, BUT IT SURE STOPS ME.
RETRY
STAHLER.
9/28
www.gocomics.com

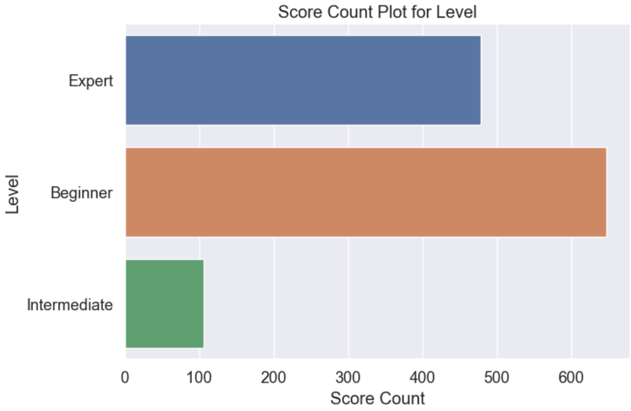
©2016 Jeff Stahler/Dist. by Universal UClick for UFS

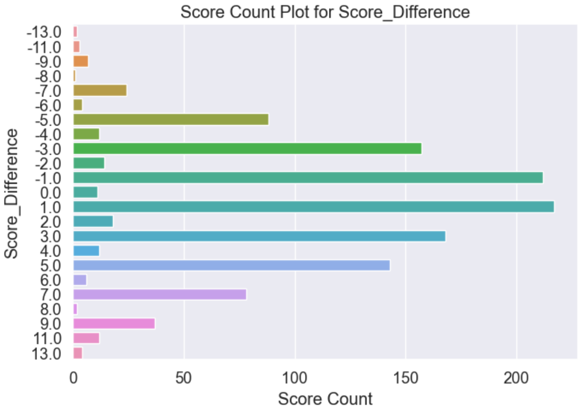
The following diagrams provide a visual representation of the score counts for attackers and defenders, as well as score counts based on the winner, level and score difference:

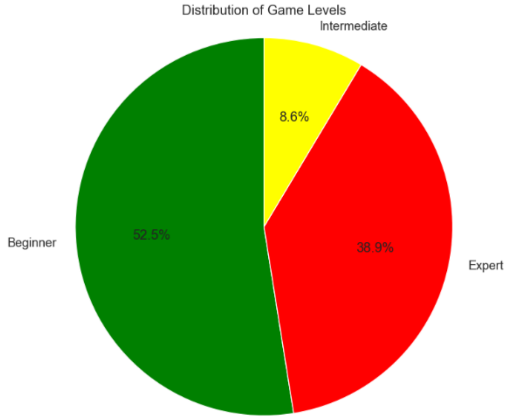






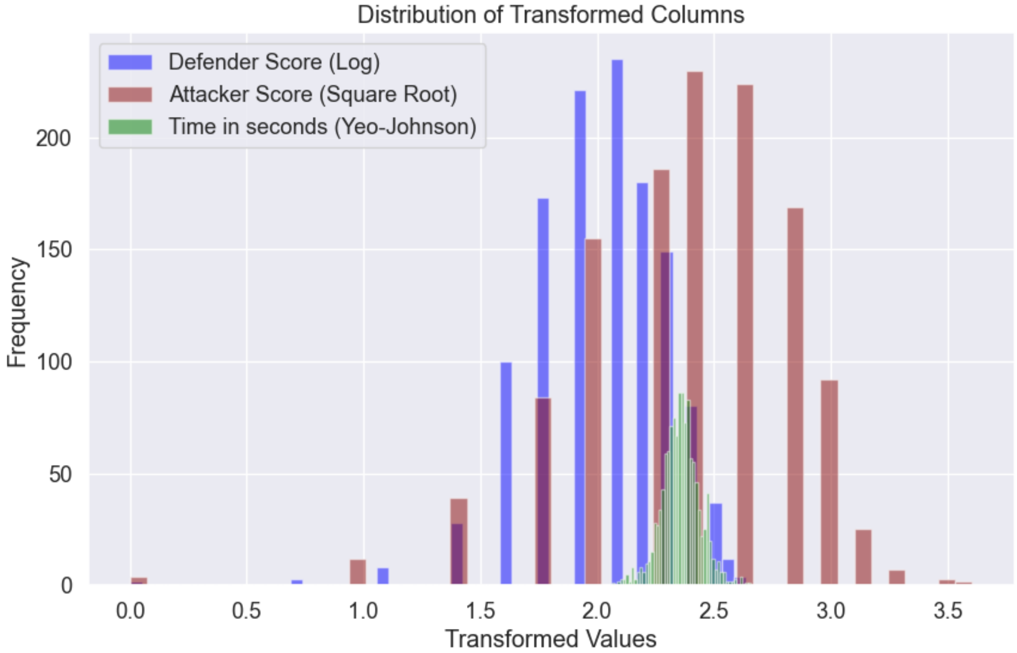




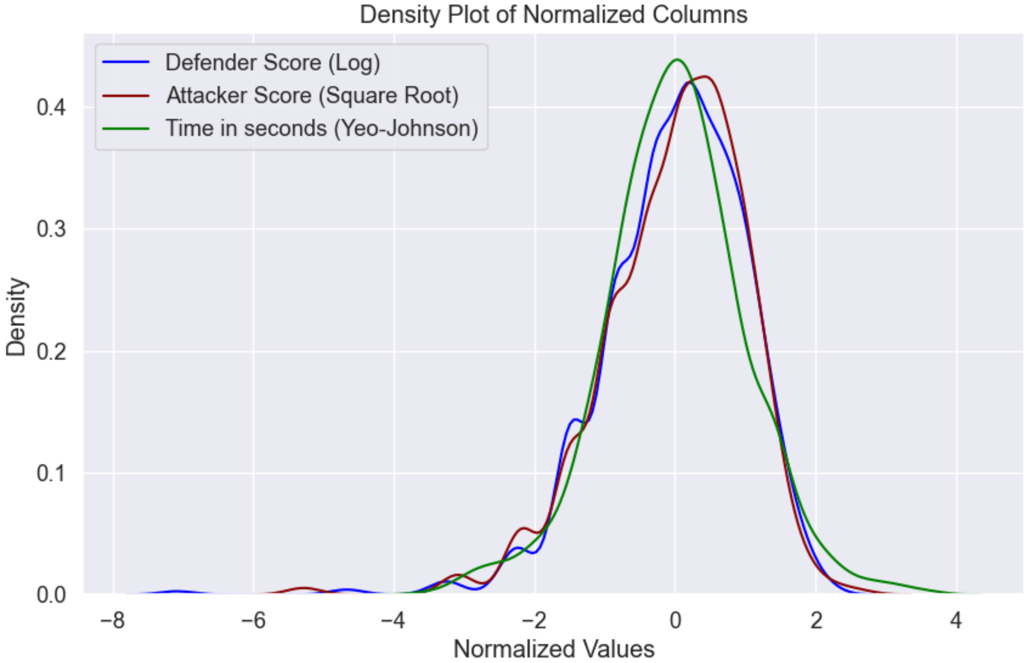


The above visualizations display that the defenders have a smaller advantage in comparison to the attackers. Although both the attackers and defenders score very similarly, indicating that the game is fair, the defenders look to have a higher score compared to the attackers. By looking into the different levels, it is evident that the players preferred to play the beginner level, as well as the expert level. This shows that the students playing the game wanted either a very easy game, or a very challenging one. By looking into the score differences between the defenders and the attackers, it shows that the score differences are very small, meaning that the all the different rounds are very competitive. The outliers for this would indicate that the student was playing the incorrect level.

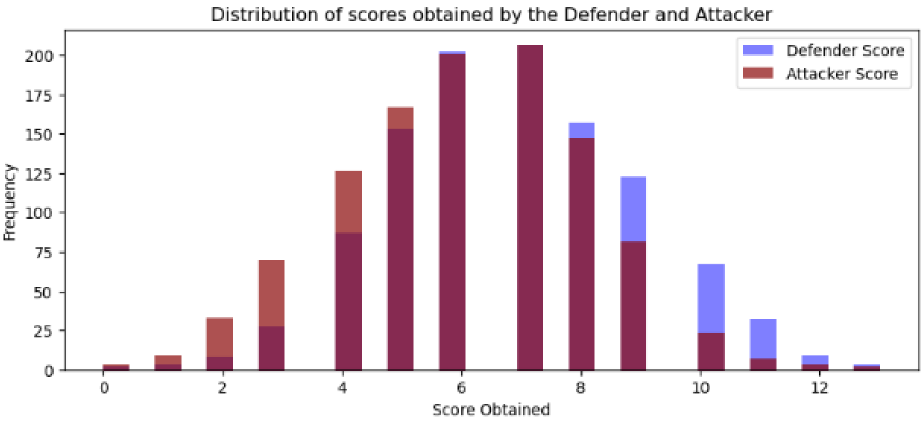
The below diagram illustrate that each transformation was implemented to allow for the data to be more normally distributed. By applying the log transformation for the Defender Score and the Square Root transformation for the Attacker Score, the skewness of each graph seems to have been reduced.

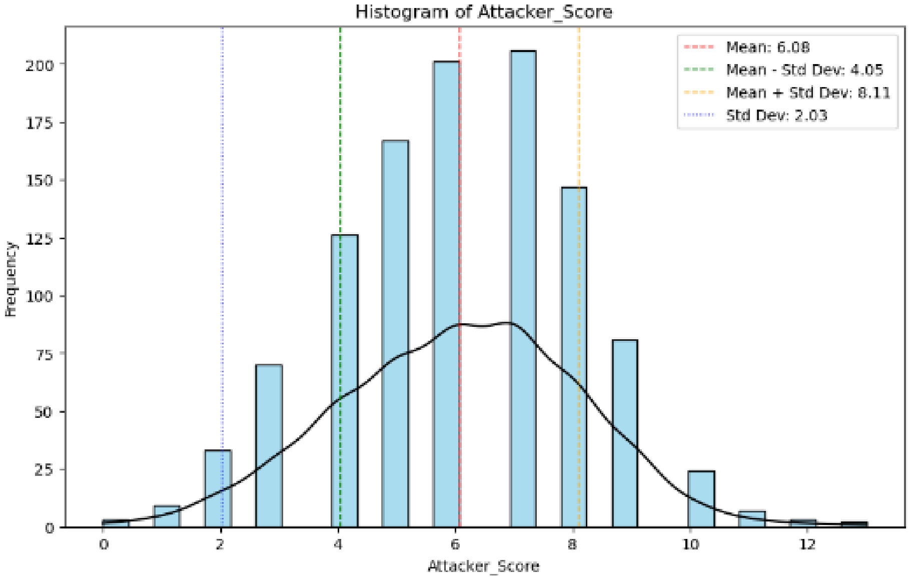


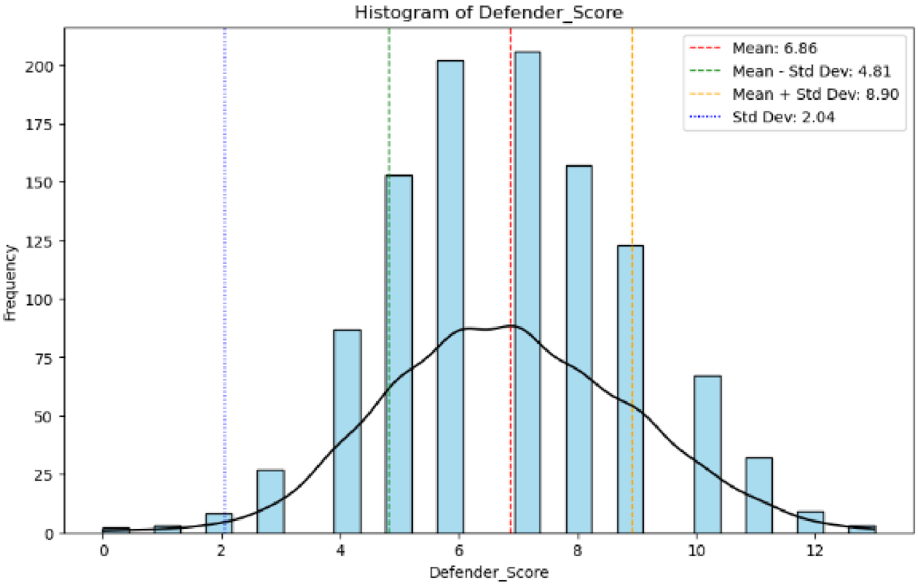
The diagram below suggests that the Defender Score, Attacker Score and Time in seconds are closer to a normal distribution:



Alongside are diagrams providing more insight into the attacker and defender. All these diagrams illustrate that most of the scores fall in the range of five to seven points, although the defenders have a slight advantage in achieving higher scores. With the standard deviations for both the attacker and defender being similar, this indicates that most of the games that were played by the students were competitive. There could be numerous reasons as to why the defenders were scoring more points, one of which could be due to the game mechanics that allow the student to have better opportunities, or the questions and answers are easier to understand to those of attackers.



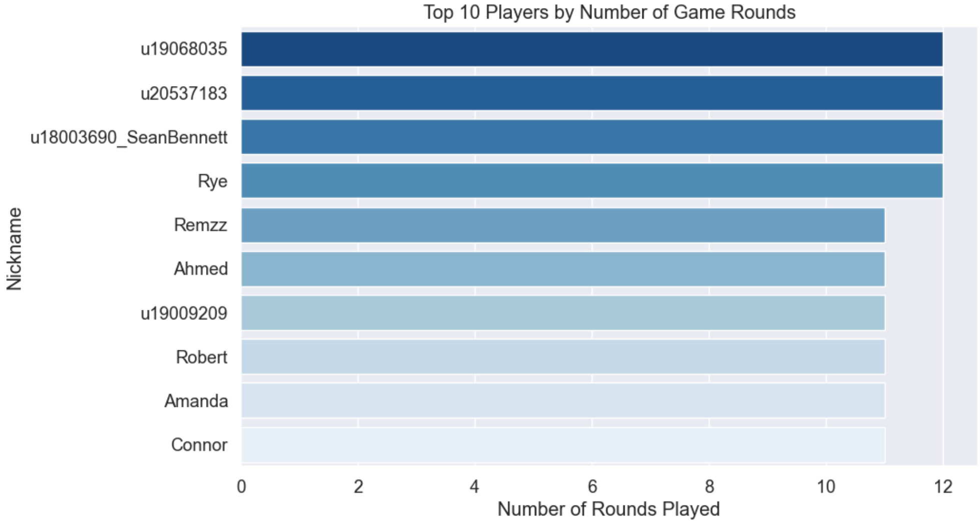


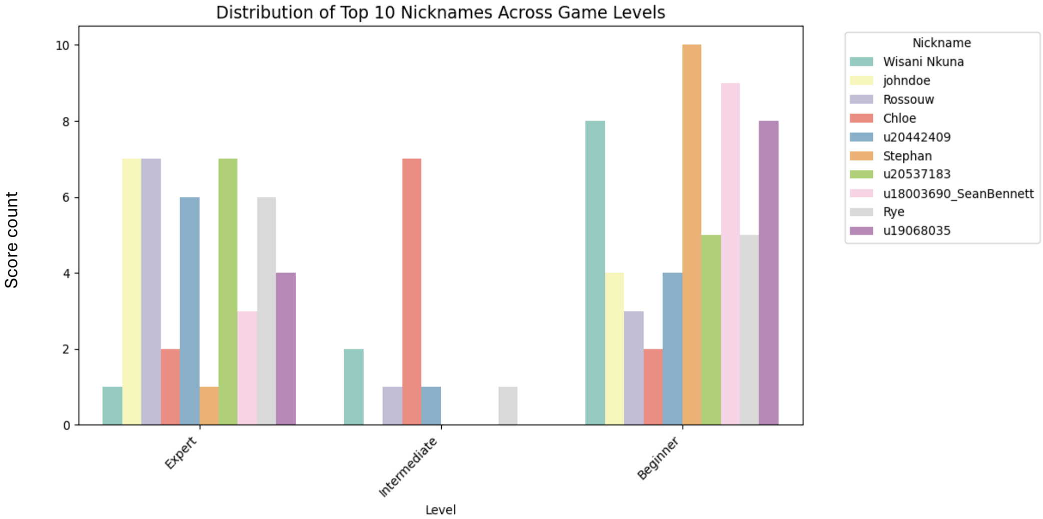


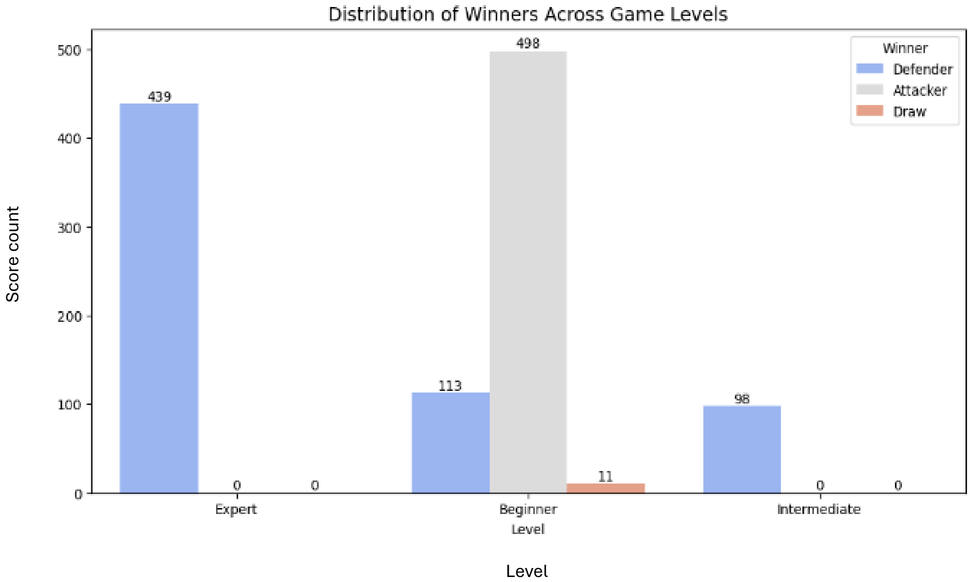
For Categorical Values

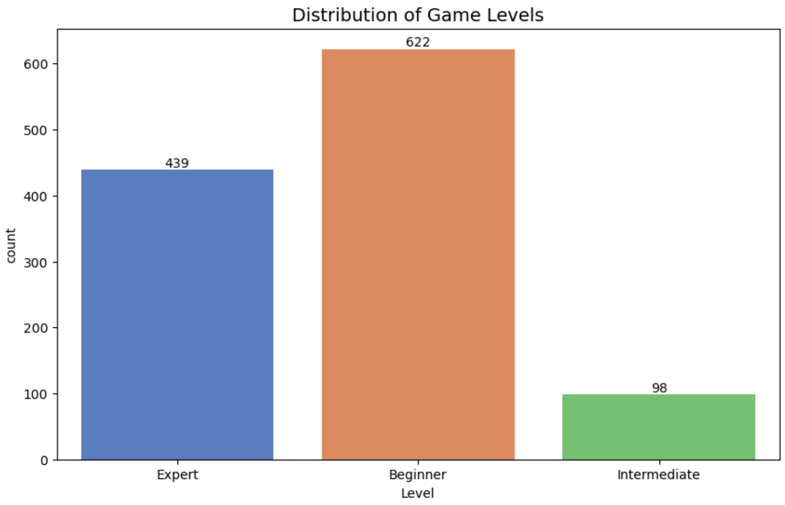
It is important to note the categorical variables in this dataset. These include Nickname, Level and Winner.

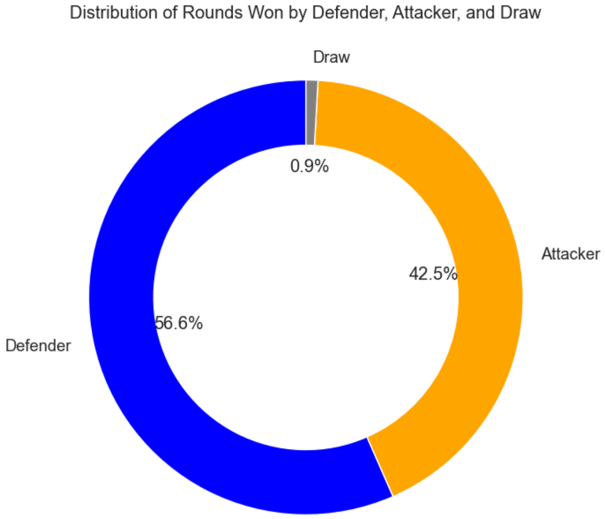
These first two visualizations display the top 10 players, who have played a similar number of rounds. In terms of the game levels, it is evident that the students preferred the Expert and beginner levels, as not too many people played the intermediate levels. This shows that the players either wanted easy or hard challenges, and not so many in between. With four players, namely u19068035, u20537183, u18003690_seanBennett and Rye, completing 12 rounds when the assignment asked for 11, will alter the variables as it shows that there are not an even number of rounds between students. Johndoe, Roussouw and u20537183 have a tie for the number of points scored for expert level. The winner for intermediate is Chloe, and the winner for Beginner is Stephan. Overall, there is a new student winner for each category, displaying that the game has various advantages to some students in certain rounds than others. The distribution of these winners across game levels in the third diagram alongside displays that the Defender is the winner for the Expert and Intermediate levels, and that the attacker is the winner for the beginner level.



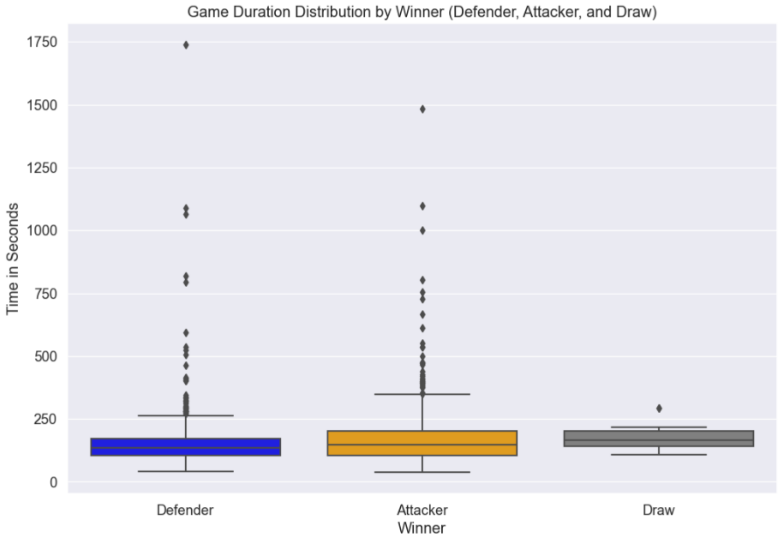


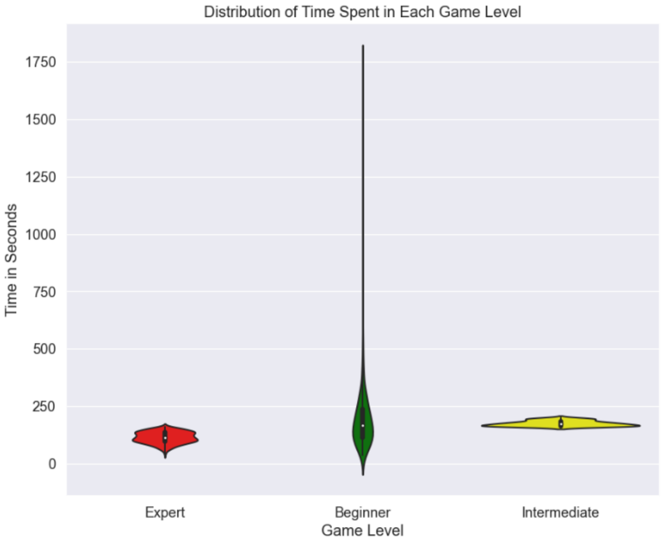






These two diagrams above illustrate that students favour the Beginner level the most, with the expert level second and the intermediate level in third. These defenders can have a ~14% advantage over the attackers in terms of the number of rounds won. Due to the very low number of draws, it is evident that the game favours a clear outcome of either a defender or an attacker, which is much better in for understanding cybersecurity, as you will always want to favour one in a situation (depending on which side you are).

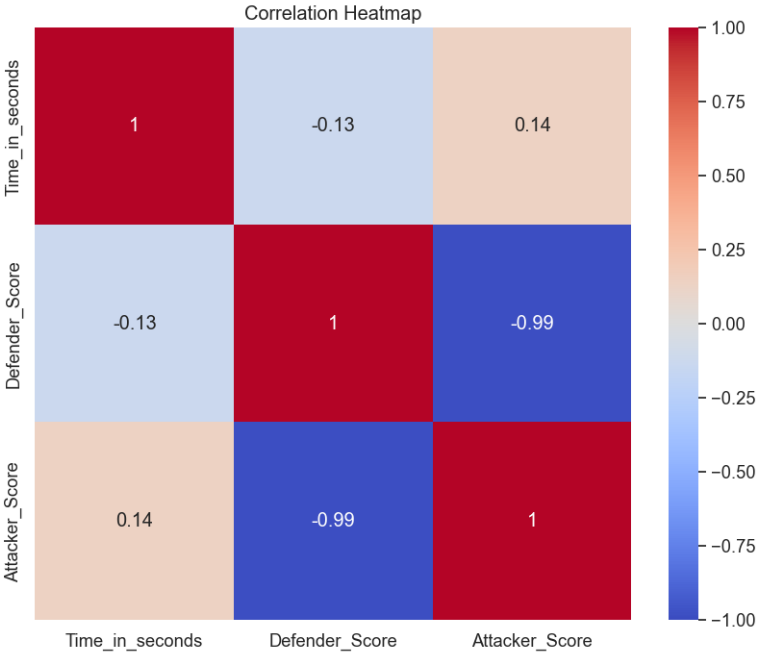




These two diagrams illustrate that the games are relatively short, with both the defenders and the attackers having a relatively similar game duration in comparison to one another. If a draw is to occur it is solved in a shorter time frame. The beginner level displays the widest variability in the game duration according to the Violin plot. This could be a display of on the diverse experience in which the players have. The intermediate and expert levels are more consistent in length. This shows that they have a more structured play for the students.

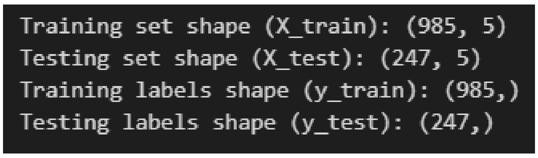
Correlation Heatmap

This correlation heatmap shows the relationship between the Attacker_Score, Defender_Score and the Time_in_seconds. These are all the numeric values in the dataset. It is evident that there is a small positive correlation between the attacker score and the time – this being 0.14. This indicates that the longer the game duration are, the weaker associated it is with attacker scores that are high. There is a weak negative correlation between the defender score and the time – being -0.13. This explains to us that the defender score decreases slightly as the game duration is increased. There is a very strong negative correlation – being -0.99 between the attacker and defender scores. This means that the if one of the roles scores increases, the other role score decreases.



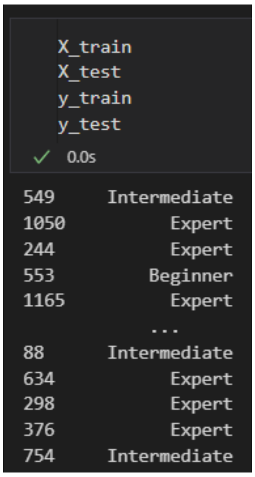
Machine Learning Implementation

To complete the machine learning, I needed to initiate the LabelEncoder and encode the categorical variables. I then needed to remove my Score Difference column and then separate the features (X) and target variables (Y). I then split the dataset into training (80%) and testing (20%) sets. My output up until this point looked like this:



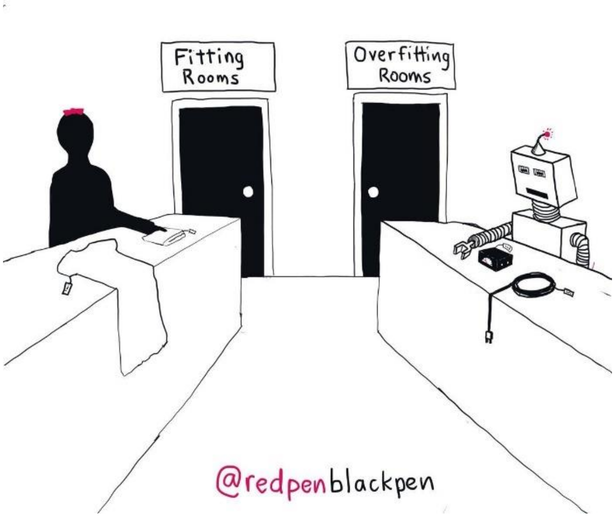
```
Training set shape (X_train): (985, 5)
Testing set shape (X_test): (247, 5)
Training labels shape (y_train): (985,)
Testing labels shape (y_test): (247,)
```

When I ran my variables, this was my output:

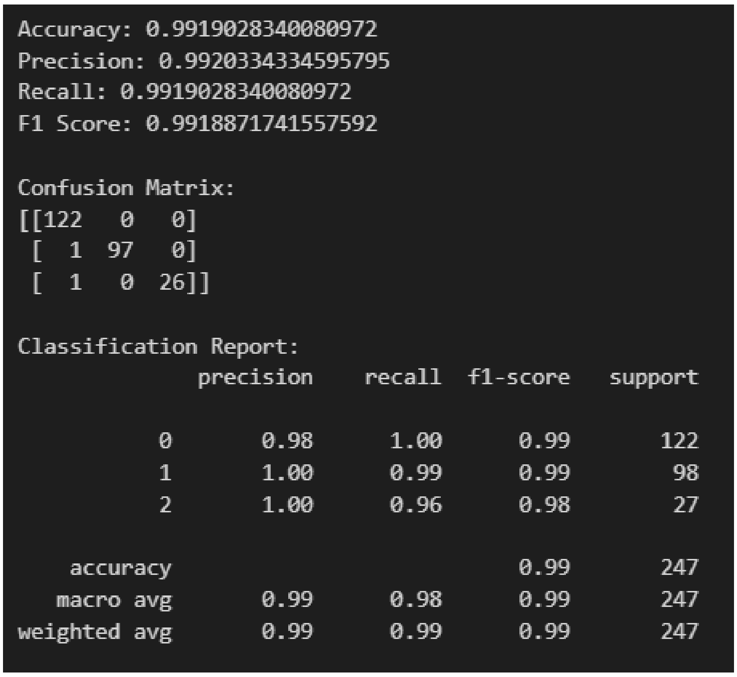


```
X_train
X_test
y_train
y_test
✓ 0.0s

549     Intermediate
1050          Expert
244           Expert
553         Beginner
1165          Expert
            ...
88      Intermediate
634           Expert
298           Expert
376           Expert
754     Intermediate
```

The output as shown here is listing the specific target labels from the testing and training sets. These will then be used for the training and the evaluating of the model's accuracy for the prediction of game levels based on what the input features are.



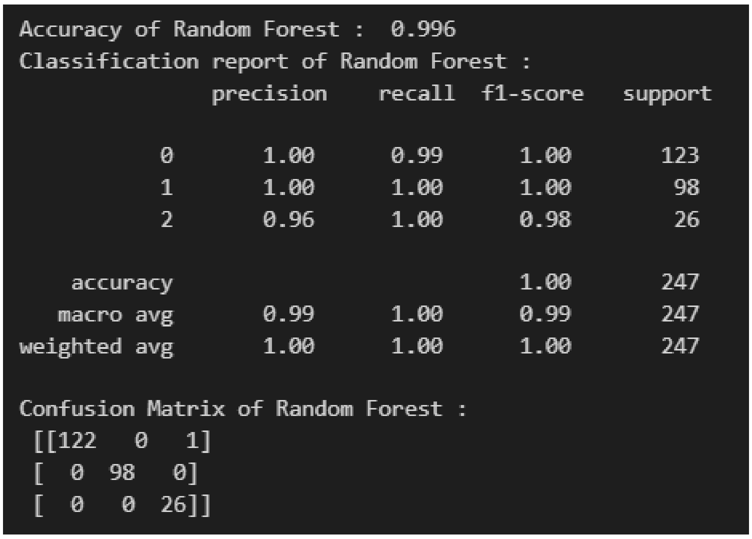
If any missing values were identified, I made sure to replace those missing values with the mean. I then initialized the models, namely the RandomForestClassifier, LinearSVC and the GaussianNB models. I then created a stacking model and fitted the stacking model on the training data. From here I was then able to make predictions on the test data and evaluate the model. My results at this point were as follows:



```
Accuracy: 0.9919028340080972
Precision: 0.9920334334595795
Recall: 0.9919028340080972
F1 Score: 0.9918871741557592

Confusion Matrix:
[[122   0   0]
 [  1  97   0]
 [  1   0  26]]

Classification Report:
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       122
           1       1.00      0.99      0.99        98
           2       1.00      0.96      0.98        27

    accuracy                           0.99       247
   macro avg       0.99      0.98      0.99       247
weighted avg       0.99      0.99      0.99       247
```

The alongside model displays the accuracy of 99.2%. This is heightened through the prediction of Beginner and Expert Levels. However, it struggles with the intermediate level. Based on the precision, recall and F1 scores scoring high amongst all the classes, this shows that the stacking classifier balances multiple algorithms for accurate predictions. The confusion matrix emphasises only some misclassifications. However, the overall models' predictive capabilities are strong.

I then completed the output for the Random Forest Model:

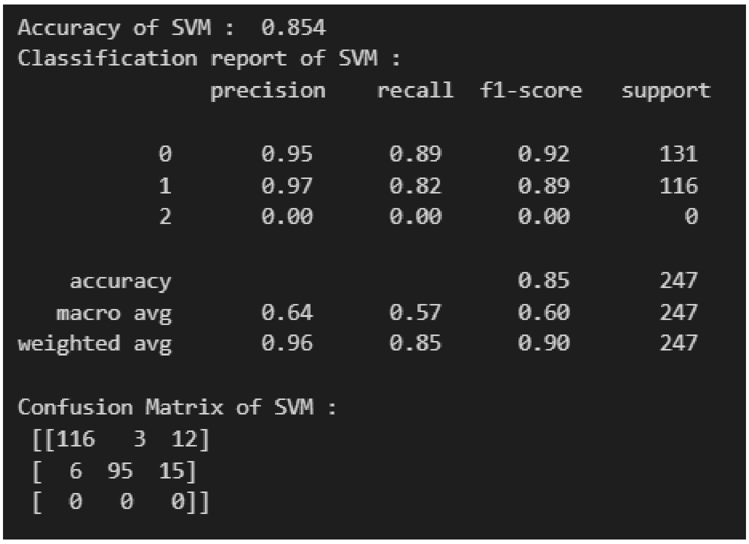


```
Accuracy of Random Forest :  0.996
Classification report of Random Forest :
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       123
           1       1.00      1.00      1.00        98
           2       0.96      1.00      0.98        26

    accuracy                           1.00       247
   macro avg       0.99      1.00      0.99       247
weighted avg       1.00      1.00      1.00       247

Confusion Matrix of Random Forest :
 [[122   0   1]
 [  0  98   0]
 [  0   0  26]]
```

The alongside Random Forest Model displays the accuracy of 99.6% in predicting game levels. Also emphasising a heighten accuracy for the beginner and expert level as viewed by the precision, recall and F1-score being 1.00. The confusion matrix displays only one misclassification within the beginner class. This overall model proves well in its results and findings.

I then made sure to do the handling of the missing data (if there was any) by inputting the mean into the cell that doesn't contain any data. I then made sure that I split the data into training (80%) and testing (20%) sets. This now LinearSVC model can then make predictions for the Accuracy of the SVM, Classification report of the SVM, as well as the confusion matrix of the SVM.

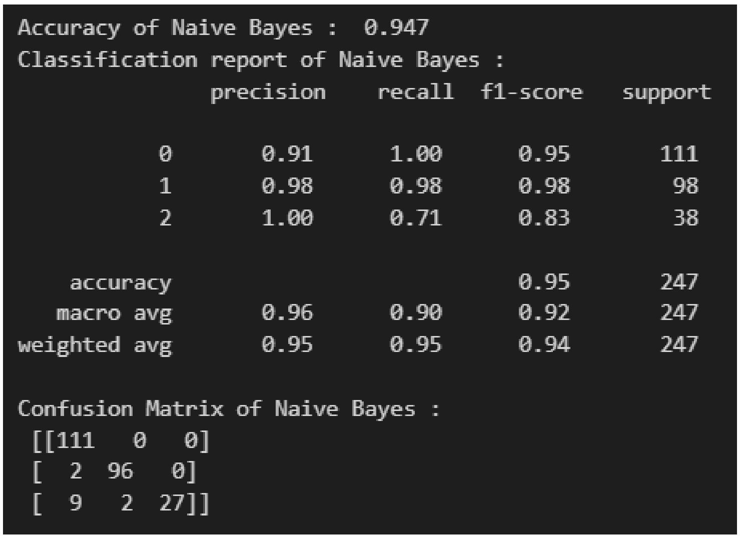
These results are displayed below:



```
Accuracy of SVM :  0.854
Classification report of SVM :
              precision    recall  f1-score   support

           0       0.95      0.89      0.92       131
           1       0.97      0.82      0.89       116
           2       0.00      0.00      0.00         0

    accuracy                           0.85       247
   macro avg       0.64      0.57      0.60       247
weighted avg       0.96      0.85      0.90       247

Confusion Matrix of SVM :
 [[116   3  12]
 [  6  95  15]
 [  0   0   0]]
```

This model displays an accuracy of 85.4%, namely with a strong performance in both the beginner and expert levels as well. The model failed to predict the intermediate class. This is backed by the 0 in the support category. The confusion matrix demonstrates that most of the misclassifications occurred between the beginner and expert levels. This is proved by 12 instances of misclassifying Beginner and expert and 15 instances of the other way around. Overall, this model seems to fight between the distinguishing of these two classes, let alone leaving out the intermediate class.

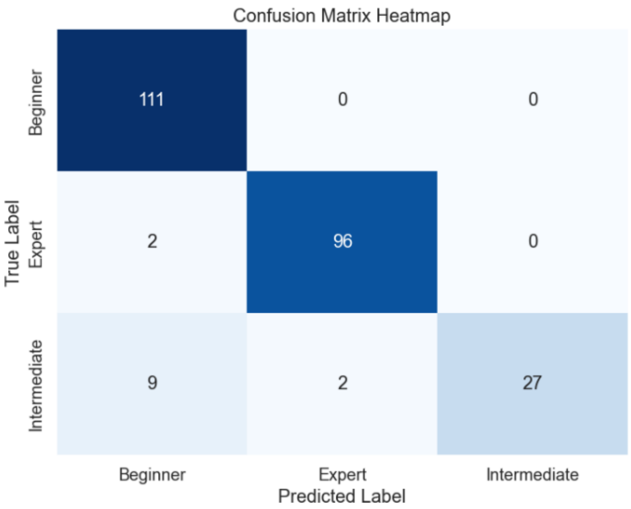
For the creation of the Naive Bayes implementation, I made sure to do the handling of the missing data (if there was any) by inputting the mean into the cell that doesn't contain any data. I then made sure that I split the data into training (80%) and testing (20%) sets. This data was then able to create predictions based on the test data. The script then evaluated the model and printed the accuracy of Naïve Bayes, the classification report of Naïve Bayes, as well as the confusion matrix of Naïve Bayes. This output can be seen below:

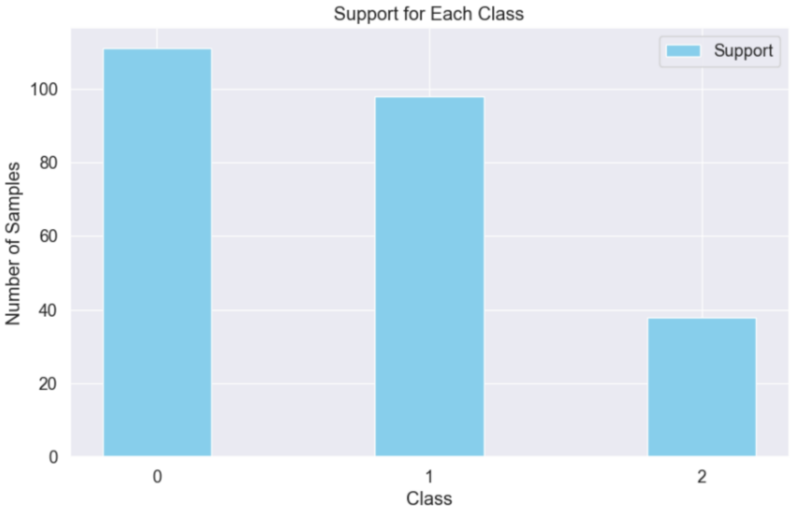


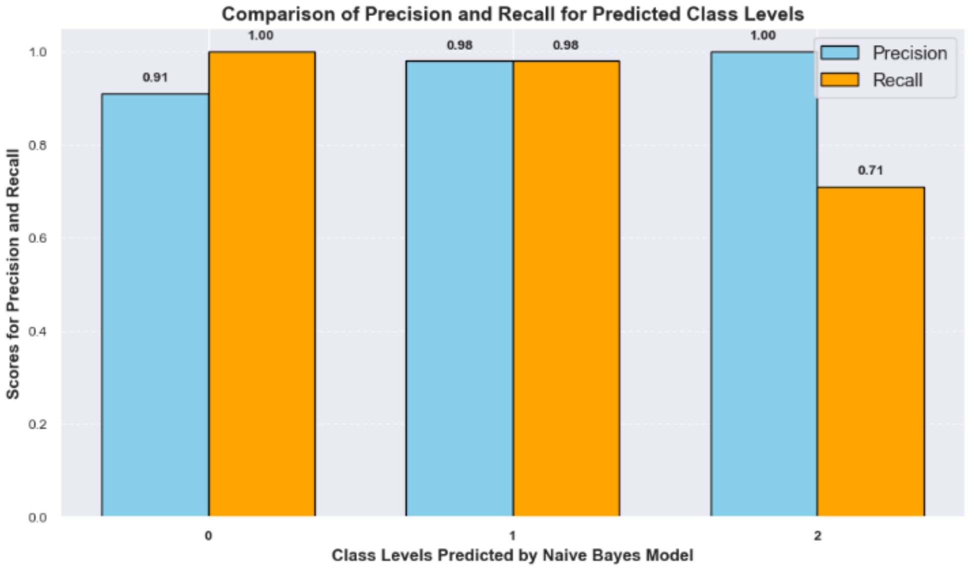
```
Accuracy of Naive Bayes :  0.947
Classification report of Naive Bayes :
              precision    recall  f1-score   support

           0       0.91      1.00      0.95       111
           1       0.98      0.98      0.98        98
           2       1.00      0.71      0.83        38

    accuracy                           0.95       247
   macro avg       0.96      0.90      0.92       247
weighted avg       0.95      0.95      0.94       247

Confusion Matrix of Naive Bayes :
 [[111   0   0]
 [  2  96   0]
 [  9   2  27]]
```

The Naïve Bayes model displays an accuracy of 94.7% and performs well over all the defined classes. The model perfectly identifies the beginner instances. This is backed by a recall rate of 0.91 and 1.00 and the F1-score being 0.95. The precision of the expert class is 0.98 with a high F1-score, meaning that it performed well. The intermediate score missed a few instances. The confusion matrix displays some misclassifications, where 9 beginner instances were incorrectly predicted as intermediate or expert levels. Ultimately the model displays a good predictive outcome.

The following graphs display the confusion matrix, support for each class based on the number of samples, and the comparison of precision and recall for predicted class levels.

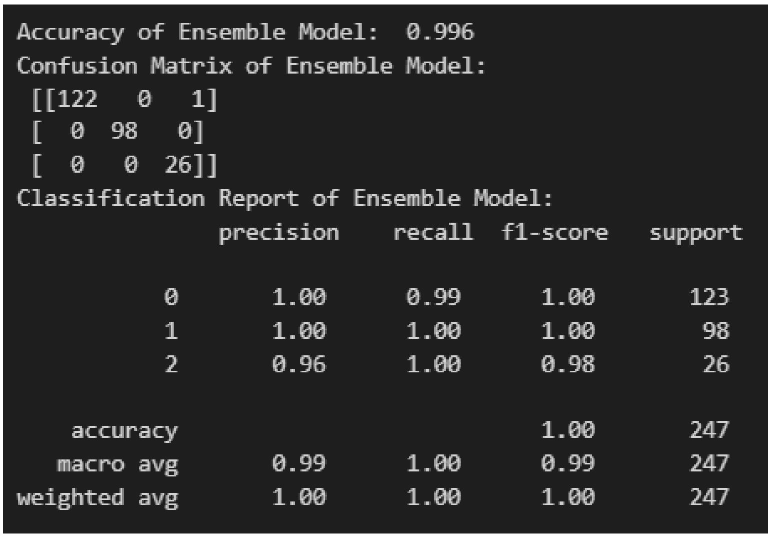




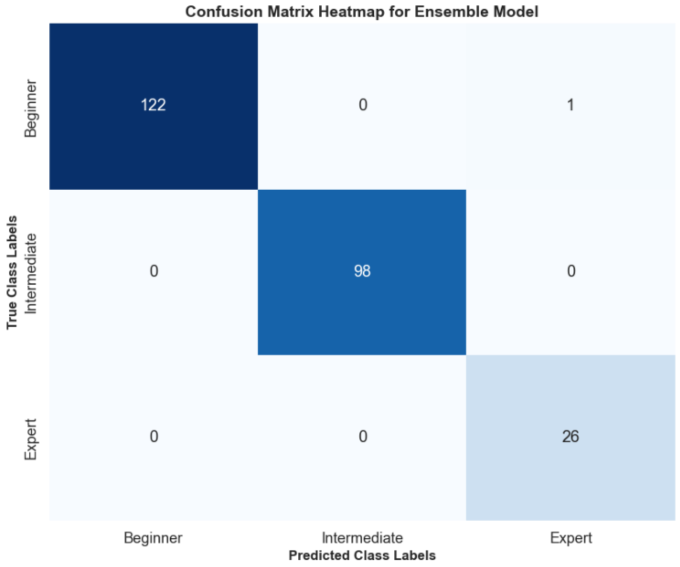


To summarize the overall 3 findings above, the model performed well for the beginner and expert levels. This is evident through the high recall and precision percentages and numbers. Unfortunately, the intermediate level class struggled. This could have been due to there being less instances of the dataset as many favoured the expert and beginner levels as mentioned before in this report.

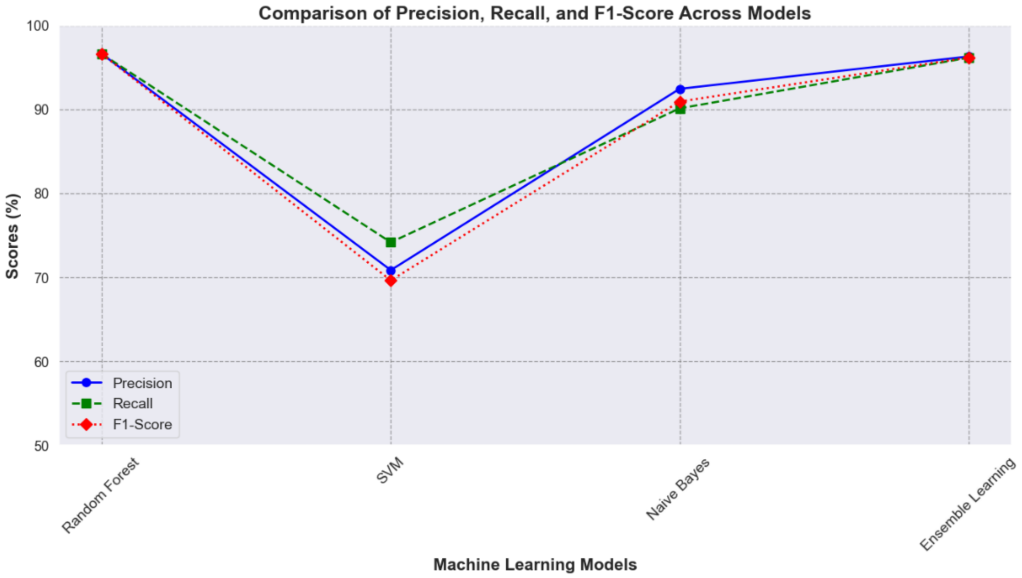
Ensemble Learning



```
Accuracy of Ensemble Model:  0.996
Confusion Matrix of Ensemble Model:
 [[122   0   1]
 [  0  98   0]
 [  0   0  26]]
Classification Report of Ensemble Model:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00       123
           1       1.00      1.00      1.00        98
           2       0.96      1.00      0.98        26

    accuracy                           1.00       247
   macro avg       0.99      1.00      0.99       247
weighted avg       1.00      1.00      1.00       247
```

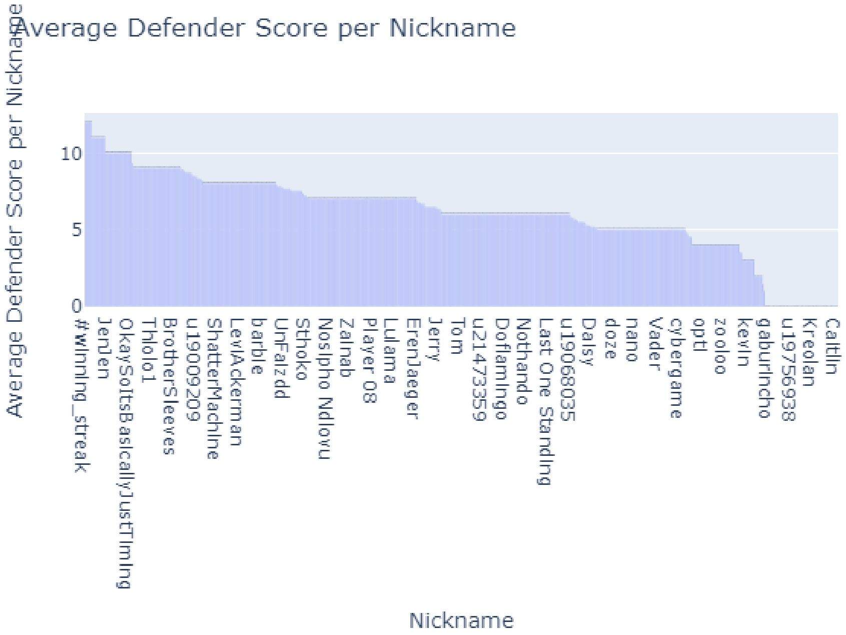


The Ensemble model received an accuracy of 99.6%. This model made 122 predictions that were correct with regards to the beginner class, 98 for the intermediate class and 26 for the expert class. Throughout this model there was only one misclassification where an expert was predicted as a beginner. The recall and precision values of 1.00 makes the model perfect. Even the intermediate class which has been a problem up until now has performed well. The recall of 1.00 and 0.96 as a precision has resulted in the model performing well. The macro average F1-Score is 0.99 and the weighted average F1-score is 1.00 displays that the model is balanced within the dataset.

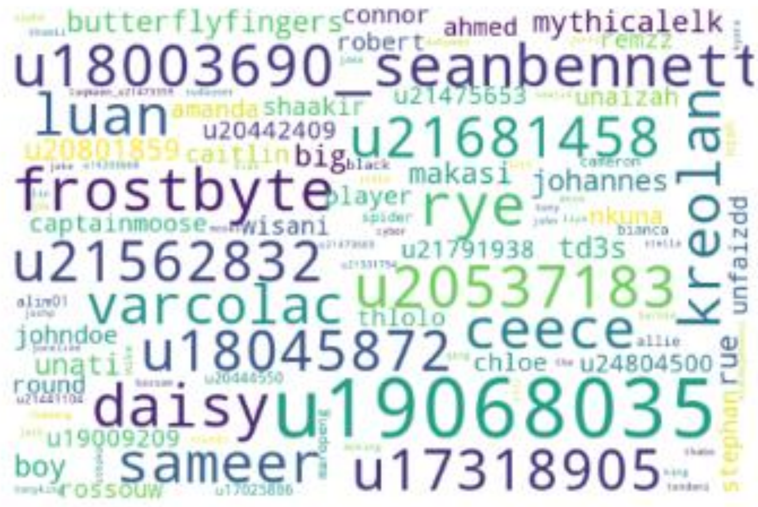


Based on the graph alongside it is shown that the Ensemble Learning model us the best model to use for this dataset. Second would be the Naïve Bayes and Random Forest. And lastly the SVM struggles to classify data.

Additional Insights to know



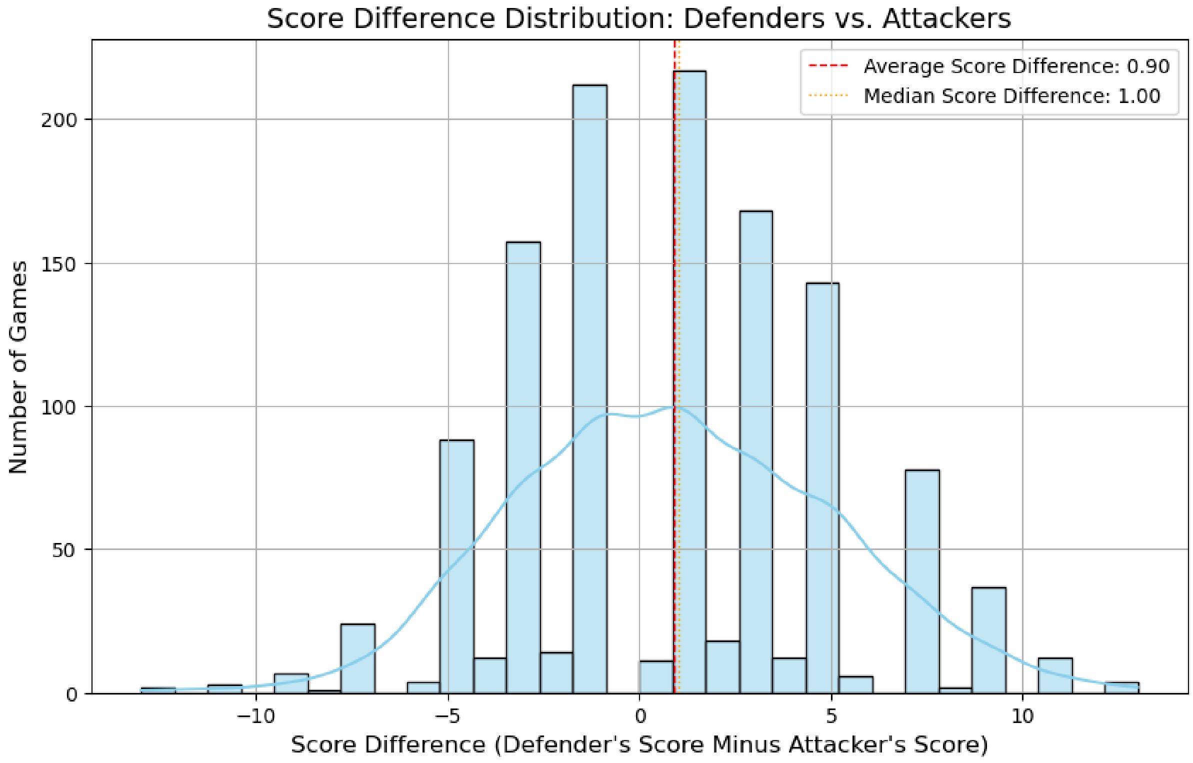
The diagram provides information into the win distribution amongst players. The visuals display the top performing players according to their average defender score where the person with the highest average is #Winning_streak followed by JenJen.

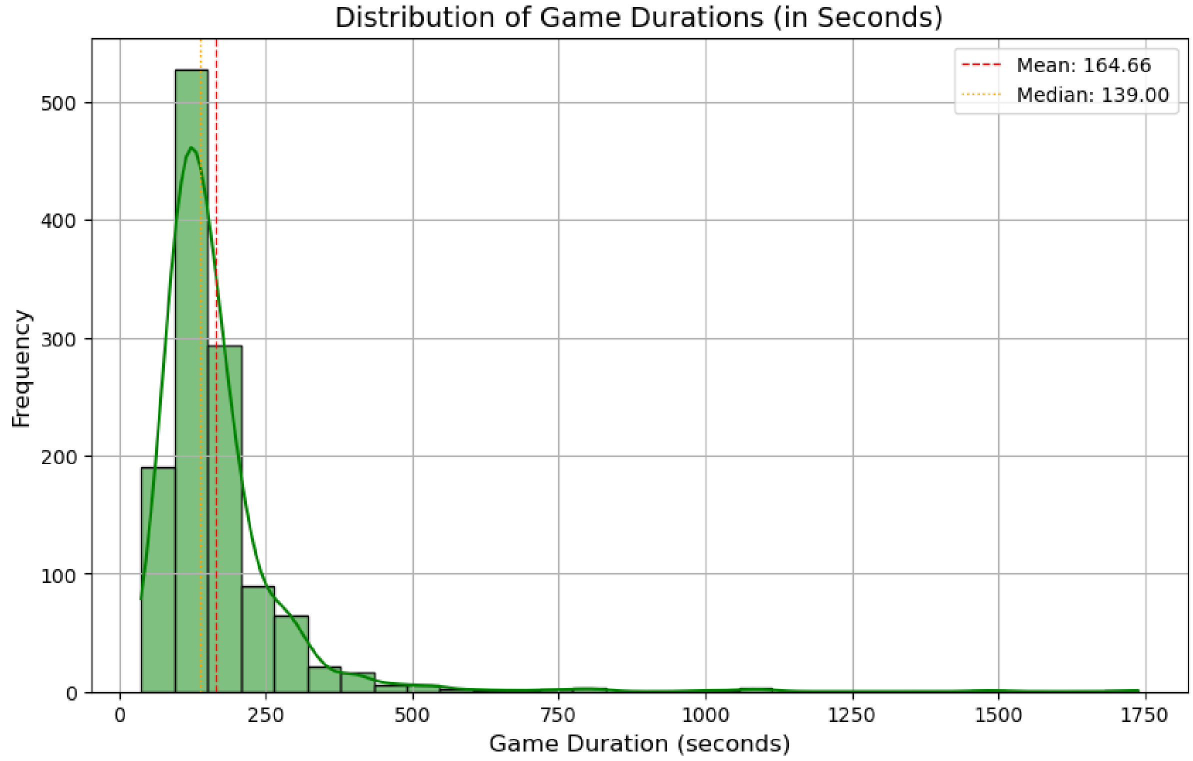


This graph represents the amount total score by each player. The bigger the name, the bigger the scores that they achieved. The top scorers are u19068035, frostbite and daisy to name a few.

Analytics Use Cases

To provide some additional insights other than the ones mentioned before with regards to the game improvements or cybersecurity enhancements:





Game Improvements:

1) There is a strong negative correlation between the defender and the attacker score (-0.99). This means that for every point a certain role receives, the other loses. To improve the balance of the game and to make it a much closer game for the attacker and the defender, a way should be developed that they each gain and lose points at the same time. This could help if the game becomes a multiplayer game.
2) The duration of the game is skewed to the shorter times. This is based on the mean being ~164 seconds. However, there are some outliers going up to 1500 seconds. By reducing the game time per round or by setting time limits for each round, would make the players more focused and engaged.

Cybersecurity Enhancements:

1) Due the competitiveness of the game, high scoring players (whether they be defenders or attackers) must be monitored for any unusual patterns. These unusual patterns could be players cheating. Any anomalies should be noted, and those specific players should be monitored.
2) Any outliers for the time aspect to complete a game (especially those that used much less time) should be monitored as this could indicate bots used to react to certain answers and questions. Any anomalies (those very short or long) should be flagged and monitored. Also introducing a time limit per round/question would rule out the outliers taking much longer than others.

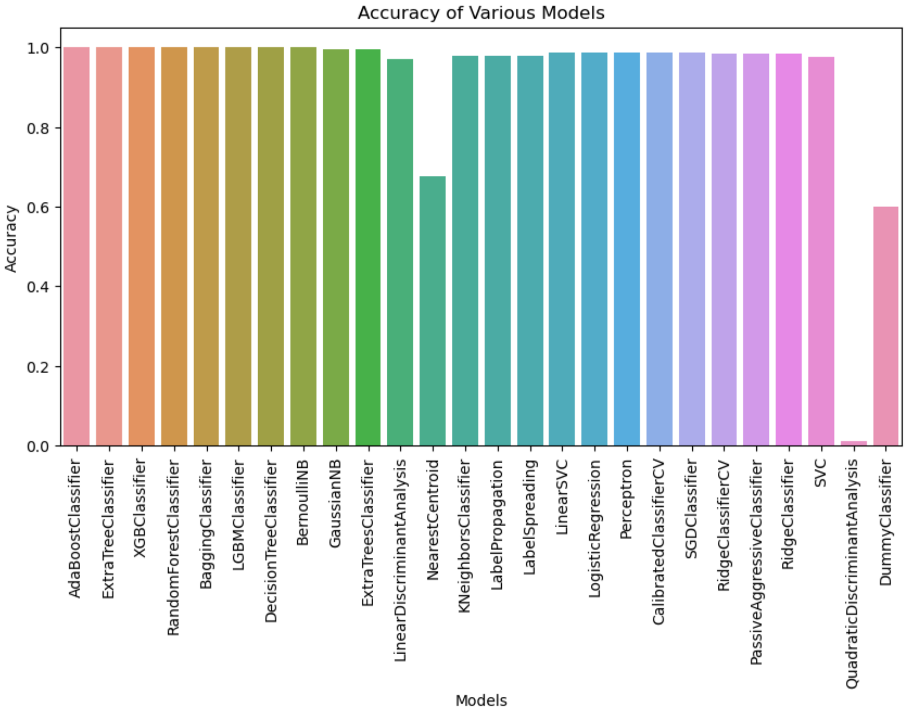
General:

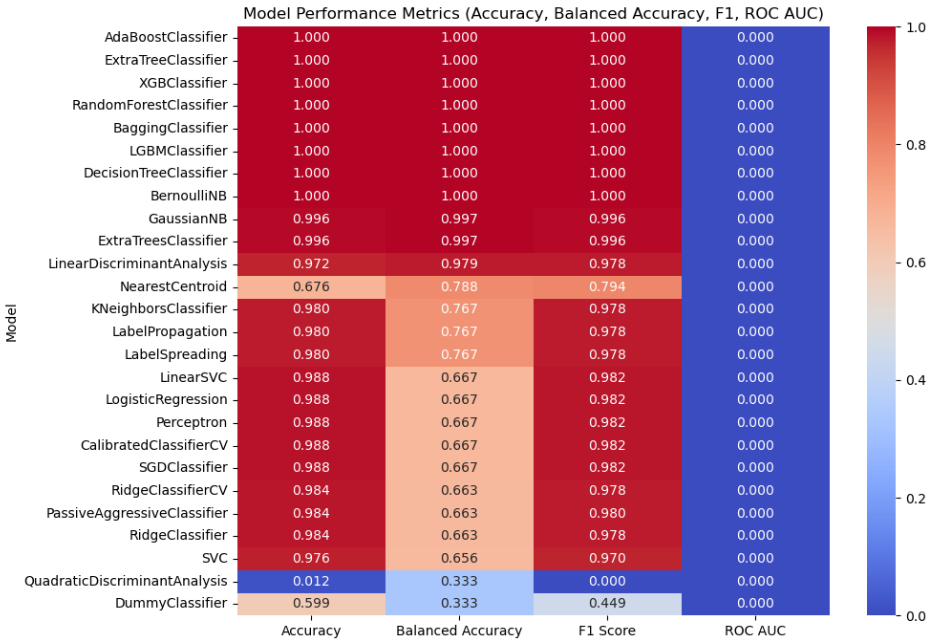
As the levels get harder it would be beneficial to make not just the questions harder, but the number of cards that the player would need to look at. Thus, at the beginner level they play with one card, intermediate has two cards and then expert has three cards.

LazyPredict

My script created the following two outputs which compared all 26 models based on the accuracy, and then a heatmap based on the accuracy, balanced accuracy, F1 and ROC AUC.

According to these visualizations on the output of using LazyPredict, it is evident that RandomForestClassifier, AdaBoostClassifier, and XGBoostClassifier are the top performing models because they are perfect with their accuracies amongst every single metric. They are perfect as they can reduce overfitting, make accurate predictions amongst different metrics, and handle complex data patterns. The worst models include the QuadraticDiscriminantAnalysis, DummyClassifier and the NearestCentroid. This is because they are not able to learn and thus make up unrealistic and not accurate assumptions about the distribution of the data.





Discussion

Data Preprocessing:

By making sure that there was sufficient preprocessing of the data resulted in the success of the machine learning models. This in turn was able to offer accurate predictions. This was done by handling missing data for example.

Game Data Analysis:

The data that was obtained displayed that although the attackers and defenders performed alike, the defenders had a slight advantage. This is good as it then displays that the game is mostly balanced.

Game Duration:

Based on the data it seems as though most of the games were short, however there were a few outliers. This could mean that either some players didn't understand or know what they were doing, or it might have been a strategic plan.

Levels:

The levels that were favoured by the students were the beginner and the expert levels. Students didn't participate in the intermediate level as much. This shows that the students ether wanted an easy game or a very challenging one.

Model Accuracy:

After viewing all the models, it is evident that the ensemble model performed the best. It had a very close perfect accuracy of 99.6% in more so the beginner and expert level. The SVM model struggled to predict the intermediate class, but most models did. This could be due to the lack of test that were done with the intermediate level.

Results of Machine Learning:

All models seemed to perform relatively well, but the SVM model struggled with misclassifications.

The insights on the gameplay:

The game itself seemed to be very competitive, with most the scores being very close for both the defender and the attacker. The game is also balanced, and this is evident from the score distribution.

Results on LazyPredict:

RandomForestClassifier, AdaBoostClassifier, and XGBoostClassifier are the top performing models because they are perfect with their accuracies amongst every single metric. The worst models include the QuadraticDiscriminantAnalysis, DummyClassifier and the NearestCentroid due to the lack of learning and the making inaccurate assumptions.

Conclusion

Summary:

The following report was able to analyze the performance of players in the cyber vigilance game and identified that the game is balanced however there is a slight advantage for the defender. According to ensemble learning models, the predictive outcomes for the game achieved an accuracy of 99.6%.

Recommendations:

Small recommendations could include that each player needs to complete an equal number of levels, thus not having an imbalance for the intermediate level's predictions and accuracies. A further recommendation could include a time limit on the duration that each player plays for, thus improving engagement and focus.

Future Work:

Future work could include multiplayer dynamics, whereby each player will get a chance to play both attacker and defender roles. Another idea in terms of machine learning techniques would be to investigate the player behavior over a much longer period.

References

Denning, T., Lerner, A., Shostack, A., & Kohno, T. (2013). Control-Alt-Hack: The design and evaluation of a card game for computer security awareness and education. *Proceedings of the 2013 …*. https://doi.org/10.1145/2508859.2516753

Maqsood, S., & Chiasson, S. (2021). Design, development, and evaluation of a cybersecurity, privacy, and digital literacy game for tweens. *ACM Transactions on Privacy and Security …*. https://doi.org/10.1145/3469821

Scholefield, S., & Shepherd, L. A. (2019). Gamification techniques for raising cyber security awareness. *HCI for Cybersecurity, Privacy and Trust: First …*. https://doi.org/10.1007/978-3-030-22351-9_13

Yasin, A., Liu, L., Li, T., Wang, J., & Zowghi, D. (2018). Design and preliminary evaluation of a cyber Security Requirements Education Game (SREG). *Information and Software …*. https://www.sciencedirect.com/science/article/pii/S0950584917301921