Advanced Strategy for MMO using Interactive Visualisation

MSc Research Project
MSc Data Analytics

Swapnil Sarda
Student ID: X17170893

School of Computing
National College of Ireland

Supervisor: Dr. Vladimir Milosavljevic

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Swapnil Sarda
Student ID:	X17170893
Programme:	MSc Data Analytics **Year:** 2018 - 2019
Module:	Research Project
Supervisor:	Vladimir Milosavljevic
Submission Due Date:	12/08/2019
Project Title:	Advanced strategy for MMO using interactive visualisation
Word Count:	6003 **Page Count** 20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ……………………………………………

Date: 12/08/2019

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Advanced strategy for MMO using interactive visualisation

Swapnil Sarda
X17170893

Abstract

Massive multiplayer online games are widely popular and fiercely competitive. Currently in-game statistics are limited to summary in a tabular format. Players invest many hours in honing their skills and have no clear view on which attributes of the game they are better at or need improvement. This study proposes to build a custom visualisation that would display the winning patterns in the game and also aid in designing a game winning strategy. Player data is fetched using PUBG API[1] and a Kaggle data set is also used in this research. For feature selection, Boruta algorithm is applied to the dataset. The frequently occurring patterns are found using Apriori algorithm. The custom visualisation is built using D3 library in JavaScript. Frequently occurring patterns are generated using Apriori algorithm with support values up to 0.998. The visualisations clearly show the different subsets found in the data. The research successfully reveals interesting patterns which can be used to win the game. This paper describes the methodology which can be extended to data sets from different games and achieve similar results, and suggests improvements which can be made to achieve higher immersion rate in the game.

1 Introduction

Data visualisation is an effective way to communicate the deep and accurate meaning of data in a graphical context to an audience. There are multiple ways to achieve it by using different visual elements like a map, a chart or a simple bar graph. These elements help us spot interesting trends, frequently occurring patterns or unwanted outliers which might be present in the data. Furthermore, data visualisation can aid in selecting an algorithm to apply to the dataset or finding missing values and false positives in the data. The speed at which the data is produced possess challenge of saving it in a readable and retrievable format. The bigger challenge being the ability to understand the data, find insights and present the findings to the various stakeholders involved. One of the well-known data mining task involves finding frequently occurring items, objects or events in data sets. These results reveal patterns like buying habits of a consumer, common keywords which can make a tweet or an article popular. In the last two decades, the area of interest has been to develop fast and efficient frequent pattern mining algorithms. The results of such algorithms is a long list of textual patterns, which makes the output difficult to comprehend. Moreover, the research conducted in past, to develop various frequent pattern visualisation techniques primarily concentrates on decision trees, cluster formations, missing values, association rules. In comparison, fewer studies have been conducted to create new frequent pattern visualisations. The tools which were designed to do so, had one similar restriction, that the data is represented in two dimensional rectangular space. Other limitation is that of clearly showing superset-subset relationship between items (for e.g., {a, b} is a subset of {a, b, c}). Both the

[1] https://documentation.pubg.com/en/index.html

limitations are overcome by the Hue Saturation Value (HSVis) (Babi *et al.*, 2018)visualiser which is implemented in this research.
In massive multiplayer online games, player statistics are an aggregation of various attributes which a player can see after a varying amount of time. This does not help a player to improve the game or enable to come up with any sort of winning strategy which is suited to their usual playing style.

2 Related Work

2.1 Need for frequent pattern visualisations

Leung, C., Irani, P. and Carmichael, C. (2008) proposed FisViz. The researchers attempted to show frequent patterns by plotting *k*-itemsets as polylines which connect to k nodes in a two-dimensional space. This enabled to spot prefix extension relationship but the polylines overlap with each other making it difficult to distinguish.

To overcome this problem, WiFIsViz is proposed by (Leung, Irani and Carmichael, 2008). The screen is divided in to two halves. The left half provides the frequency information while right half lists the frequent patterns found in the dataset in the form of a tree.

Another visualisation, namely, FpVAT (Leung and Carmichael, 2010) is proposed to mitigate the short comings of FisViz. Similar to WiFIsViz, all frequent patterns and their frequencies on one single screen as a wiring-type diagram.

The above three visualisations are suitable for a dataset which does not create many frequent patterns. Thus making them unsuitable for big data. (Carmichael and Leung, 2010) designed CloseViz to tackle the big data problem. In essence CloseViz is an extension to WiFisViz and FpVAT. Instead of showing all the patterns, only 'closed' patterns are selected. A pattern Z is closed, if the frequency of Z does not match its superset.

Leung, Jiang and Irani, (2011) have created a method for visualisation of frequent patterns. Their proposed work was FpMapViz which was based on tree map visualisation for hierarchical data. The patterns are generated in rectangles. Recurring item for a single rectangle is visualized within it. Although it utilizes the display area, it is difficult to comprehend and may confuse the user.

RadialViz (Leung and Jiang, 2012) is designed with the aim to build an orientation free visualiser similar to FpMapViz. RadialViz shows the frequently occurring patterns in a hierarchical way. In the radial layout, the frequency of the patterns is shown by the radius and the cardinality of the pattern is represented by the sector of the colours.

Jayaprada *et al.*, (2014), have proposed VizSFP, which uses patterns in the database which have semantically similar patterns in a dynamic database. The patterns are classified on a scale of C1 to C5. The clusters are visualised using different graphs, having different purpose. The visualisations can be more compact.

Another visualisation PyramidViz (Leung et al., 2016) uses a design different than the other visualisation. The visualisation is built as a block one on top of other. The lowest laying block shows single item frequent set. The limitation of this approach is the inability to find which patterns are more frequent than others.
G *et al.*, (2016) proposed a pre-CALI to visualise the complex protein ligand interaction (PLI). Interactions between proteins and ligands are separated by colour. Although the methodology is effective in showing frequent sub graphs that occur in the interaction, research study is limited to molecules visualisation.

Bikakis and Sellis, (2016) carried out a state of art survey to find frequent pattern visualisations. Researchers concluded that the focus of the visualisation has been on static data. More than twenty techniques have been described but not many visualisations are focused on frequent pattern mining. SemLens is one of the visualisation that can display the frequent patterns as a scatterplot. This limits the type of the data that can be used.

This research study is based on the HSVis visualiser (Barkwell *et al.*, 2018). This visualiser is orientation free and can handle large datasets as well. The visualiser represents the itemsets of different cardinalities based on colours. Instead of the radius, the visualiser divides the sector in to degree zones. The outermost ring is represented by red colour and will show the itemset of cardinality one. The frequency depends on the shade of the colour. Darker the shade, more frequently appearing is the pattern.

2.2 Frequent pattern mining algorithms and evaluation measure

2.2.1 Frequent pattern mining algorithms

Borgelt and Kruse, (2011) in his research, implements Apriori algorithm with the aim of reducing execution time. The experimentation shows the dynamic node structure leads to minimal memory usage but at a loss of performance and organizing of transactions in a prefix tree is beneficial, outweighing the cost of construction of the prefix tree.
The above implementation is part of *arules*[2] packages in R and provides ease in parameter tuning and providing evaluation measures for the patterns which are generated.

Few studies have been completed in the past to overview the developments in the field of frequent pattern mining. (Jamsheela and Raju, 2015), researchers conducted a survey to detail the new frequent pattern mining algorithms like Linear prefix tree (LP-tree), FUET, IFP growth and an adaptive algorithm proposed by the researchers. Comparison is made between execution times of different algorithms concluding, state of art are algorithms being better than basic FP growth.

In their research, (Dharani *et al.*, 2016) have used frequent pattern mining algorithms business insight and proposed a partition algorithm. The apportioning of information is significant when there are high number of rows. There are various segments in the dataset having different characteristics like names, attributes like storage and record index. To discover the patterns, database is examined just once. At that point the tally of various

[2] https://www.rdocumentation.org/packages/arules/versions/1.6-3

components is received from the transactional table. However, there is no remark on the proficiency of the algorithm on size of datasets.

Babi, (2018) recently completed a survey on state of the art pattern mining algorithms, giving an outline on an exhibition correlation of different calculations. Researchers enlist advantages and limitations of number of the mining algorithms. The related works section is orchestrated with the end goal that, each new algorithm proposed is an improvement over their very own separate forerunners. Key point to detract from these study is to utilize the algorithm which will have higher exactness when discovering frequent samples in a dataset. One of the techniques in this exploration overview, is a novel algorithm called BitTableFI for mining of successive itemset and how it is effective then other approaches.

Both the surveys on the cutting edge frequent pattern mining algorithms have inferred that every algorithm have some sort of drawback and which algorithm to utilize relies upon the size of the database and our necessities of the proposed outcome.

Heaton, (2016) has studied characteristics of a dataset which affect the performance of popular frequent pattern mining algorithms like Éclat, Apriori and FP-Growth. Dataset characteristics which have an effect on the performance are data density, which is the percentage of baskets that are intentionally present frequent itemsets. Another dataset characteristic is basket size. Comparing basket size performance, Apriori was slightly outperformed by the other two algorithms.

2.2.2 Evaluation measure

For the frequently generated patterns, their accuracy is measure as 'interest' in the rules that are mined. Lift, support and confidence are commonly known measures to determine if the rules is interesting.

Brin *et al.*, (2005) proposed Conviction as a measure for rules. Conviction is the ratio of the expected frequency that an itemset X occurs without Y, if X and Y were independently divided by the observed frequency.

Li Min Fu and Shortliffe, (2000) An increase in the certainty factor is the decrease of the probability that Y is not in a transaction with X. Vice versa for X and Y also stands true. The range for the CF is between -1 to 1. Certainty factor of 0 means that LHS and RHS are independent.

Ochin, Kumar and Joshi, (2016) proposed a measure Rule Power Factor, which accurately classifies a rule as interesting, even when some other measures like support, confidence, chi square statistic might fail, the RPF will correctly determine the interest measure of a rule. For the research purpose, RPF will be used for evaluation.

2.3 Online game data mining

In (Siqueira *et al.*, 2018), game data mining was conducted for the popular game *World of Warcraft*. The researchers in this study applied multiple linear regression to study if the player will stop playing the game in near future. Another task undertaken in this research is of profiling of the players. Multiple linear regression analysis revealed that players with higher levels in the game and high number of hours are less likely to leave the game. Player profiling is done using k-means clustering which identified four types of players. Although this research did not reveal any techniques that can help player or make game more immersive.

(Braun *et al.*, 2017), researchers analysed a team based survival game called Overwatch. There were limitations to collect data for this game, as no official API is available. The researchers carried out analysis for 1 of the hero out of more than 100 available and how to play that particular hero to maximize chances of winning the game. The technique implemented for clustering and pattern finding is affinity propagation and evaluation of the characteristics is done by plotting graphs when win rate cluster is dense.

The attempt in this research study is to design a game winning strategy and show it as an interactive visualisation, which shows patterns generated by using modified Apriori algorithm based on Borgelt's version which is updated in May 2016. The algorithm focuses on speeding up execution times of the Apriori. The evaluation measure used to determine the 'interest' in a pattern is Rule Power Factor, which is an alternative to support and confidence values.

3 Research Methodology

From related works section, we can conclude that more research is required in the field of designing of visualisation for frequent patterns found in the data. In this research study, CRISP-DM process is followed as it builds a foundation on how to conduct implementation of this research.

3.1 Business Understanding

The aim of the first phase is to select and recognize the business objective of the research. This research is undertaken to improve the gaming experience of the players, especially in massive multiplayer online games. Thus the business objective of this research is providing additional statistics to the players which will help them improve their game as an in-app purchase. Alternatively, developers can also interpret player data of every single player and offer customized packages to the individual players.
Second objective of this phase is to describe the contributions which this research is making towards the visual analytics and online game data domains. This research makes improvements on the HSVis visualiser, making it even more easy to interpret. The methodology of this research can be extended to similar games as the one used in the research, a generic code that can be utilised by others.

3.2 Data Understanding

This phase focuses on understanding of the data used in the research. There are many games which can be classified as strategy based games for battle royale mode. The characteristics of these games is that it can be played as player vs player game or team vs team game. The objective is to win the game by eliminating other players. The dataset that is used for this research is the most popular game on the Steam platform, namely PlayerUnited's Battleground (PUBG) allows up to 130 players in a match.

Data is collected using PUBG API and also a Kaggle Dataset which include match and player statistics is used in the research. In total there are 18 attributes in the dataset and combined rows of both the dataset contain more than million rows.
The attributes in the dataset are as below.

Serial No.	Attribute name	Min	Max	Details
1	WinPlace	0	130	This player's placement in the match
2	DBNOs	0	NA	Number of enemy players knocked
3	Assists	0	128	Number of enemy players this player damaged that were killed by teammates
4	Boosts	0	NA	Number of boost items used
5	DamageDealt	0	NA	Total damage dealt. Note: Self-inflicted damage is subtracted
6	HeadshotKills	0	129	Number of enemy players killed with headshots
7	Heals	0	NA	Number of healing items used
8	KillPlace	0	130	This player's rank in the match based on kills
9	KillStreaks	0	NA	Total number of kill streaks
10	Kills	0	129	Number of enemy players killed
11	LongestKill	0	NA	Enemy killed by at longest distance
12	Revives	0	NA	Number of times this player revived teammates
13	RideDistance	0	NA	Total distance travelled in vehicles measured in meters
14	SwimDistance	0	NA	Total distance travelled while swimming measured in meters
15	TeamKills	0	NA	Number of times this player killed a teammate
16	VehicleDestroys	0	NA	Number of vehicles destroyed
17	WalkDistance	0	NA	Total distance travelled on foot measured in meters
18	WeaponsAcquired	0	NA	Number of weapons picked up

Table 1: Dataset attribute table

3.3 Feature selection

A single game consists of many attributes that may affect the outcome of the match. An assumption is made that some attributes may not play a role in winning the game. To test this assumption a feature selection test is required. For feature selection, Boruta algorithm is applied to the dataset.
Boruta algorithm is based on random forest feature selection methodology. Boruta algorithm provides unbiased and stable selection of important and non-important attributes from a data set. The iterative construction of the Boruta algorithm helps in dealing with interaction

between varying characteristics of a dataset like different types of variable type such as numeric, character or factor, and, the importance measure of random forest. The dataset for this research contains varying data types.
Boruta algorithm is a wrapper for random forest feature selection method and part of the Boruta package in R. For this research, the importance of attributes is measured against *Win Place* which shows the which attributes are contributing to win the game.

3.4 Modified Apriori algorithm

A major task undertaken in this research is to find frequent patterns in a dataset. Apriori algorithm is used to this end. The purpose of finding frequent patterns is to help in determining next set of actions with a high probability, on the basis of presence of certain objects in a given set.
The *Arules* package in R provides the Apriori function which can be used to mine the frequently occurring patterns in the dataset. The Apriori function is based on (Borgelt and Kruse, 2011) work, which has one major difference, that on RHS only single item is present which minimizes the execution time. This Apriori function provides the flexibility to explicitly describe the target type for mining rules. The target type for this research data set is defined as “frequent itemsets”. The support value is also defined as a parameter to the function. Minimum support value or $supp_{min}$ is the statistical measure to find the values with large enough basis in the data. The default value of minimum support is 10% and can be adjusted as per the needs of the research. Support value can be calculated as demonstrated below:

$$\boxed{\operatorname{sup}\left(X = \frac{|\{\, t \in T; X \in t\}|}{|T|}\right)}$$

Figure 1: Support value calculation

But keeping the minimum support to default will increase the number of patterns that are generated and the patterns may very well exceed the entries of transaction matrix.
To perform the Apriori algorithm, the data frame needs to be coerced to a transaction data type. Every item group have a unique id in transaction data.

Since every layer of the visualisation represents different set of items, the output of the Apriori algorithm is subset in six different files. First file will be subset of frequent itemsets of cardinality one, second will be a subset having frequent itemsets of cardinality two and so on.

Every new file that is a subset is feed as an input to the visualisation. Outermost donut chart will have input of subset with itemset cardinality of one, working towards inner donuts and assigning them files with sequential increasing set of cardinality. The circle will have input of the subset with itemset cardinality six.

3.5 HSVis visualiser

Hue Saturation Value visualiser is based on the works of 2018 Manitoba students. The result of the Apriori algorithm implemented using *Arules* package is given as input to the HSVis. As observed in related works area, very few models focus on the direction of the visualisations. In the research work previously conducted, the visualisations are confined to a space, constraining the measure of data that can be understood like superset-subset relationship.

A model which has these restrictions is FpMapViz (Leung, Jiang and Irani, 2011). HSVis utilizes a roundabout and direction free design. This model optimizes the usage of hue of different colours. Every colour symbolizes itemsets with different cardinality.

Degree	Cardinality	Colour
0°	1 item itemset	Red
60 °	2 items itemset	Yellow
120 °	3 items itemset	Green
180 °	4 items itemset	Blue
240 °	5 items itemset	Cyan
300 °	6 items itemset	Purple

Table 2: Cardinality table

Anomalies or spikes in the visualisation is handled by the saturation of the colours and the values it represents. Saturation lies in the range of 0.5 and 1. The higher the saturation, the tint appears to be darker. Alternatively, if tint is whiter in shade, saturation tends to be 0.5. If 'Value' is close to 1, the tint gives off an impression of being light and when Value is 0.5, shading seem, by all accounts, to be darker.
The 'Value' here represents the support values that are found in the frequently occurring patterns result.
To summarize, the darker the colour tint, higher is the probability of that pattern will yield similar results as seen in the data.

	S = 0.5, V = 1 Less frequent		**S Increasing V Decreasing**		**S = 1, V = 0.5 Most frequent**
H = 300°					
H = 240 °					
H = 180 °					
H = 120 °					
H = 60 °					
H = 0 °					

Table 3: Saturation and Value table

When the saturation and value tend to be equal the colour tends to be grey which will prove to be difficult to read. Therefore, the minimum values of saturation and values are kept 0.5 so that there will always be some difference between the two values and they will not appear as grey.

The visualisation is designed using D3 library in JavaScript. There are some changes made to the visualisation which provides finer control over it.

First is the addition of the legend, which corresponds to each subset. The legends act as an "on-click" event. The visualisation of only selected subset will be visible on the screen. To reset the view of other subsets, again an "on-click" event is bind to the particular subset legend. The legend works as a toggle switch to turn on and off the visibility of the remaining subsets in the visualisation. The reason behind doing this is to make the individual subsets more accessible.

Second improvement is the animation and expand effect. To clearly view this, we first focus on a single subset. Then clicking on any slice will rotate the selected slice to the 90° position on the circle. The aim was to add the elegance factor to the visualisation.
Third improvement is the addition of tooltip. The tooltip is displayed on hover. The position of the tooltip is fixed. Tooltip displays the frequent itemset label, as well as the data that has been occurring frequently.

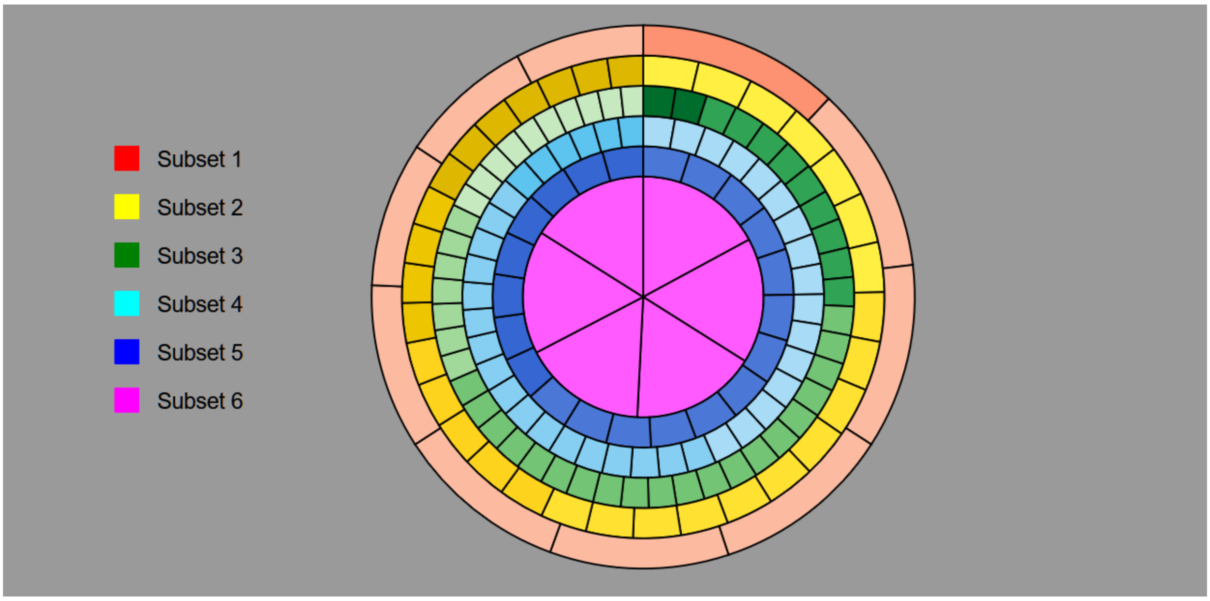


Figure 2: Hue Saturation Value(HSVis) visualiser

4 Design Specification

First step is to retrieve data from PUBG API and load the Kaggle dataset in to the environment. Once the data is loaded, apply feature selection technique Boruta algorithm to find which attributes are important in winning the game. Then modify the dataset and discard the unimportant features. Next step is to bin the data set in six equal bins and convert the data type to transactions. Then apply modified Apriori algorithm using the Arules package in R on both the datasets. Evaluate sets of rules generated using RPF as measure and compare the median support values. Select the dataset to visualise and supply it as input to the HSVis visualiser. Analyse the visualisations and design a game winning strategy.

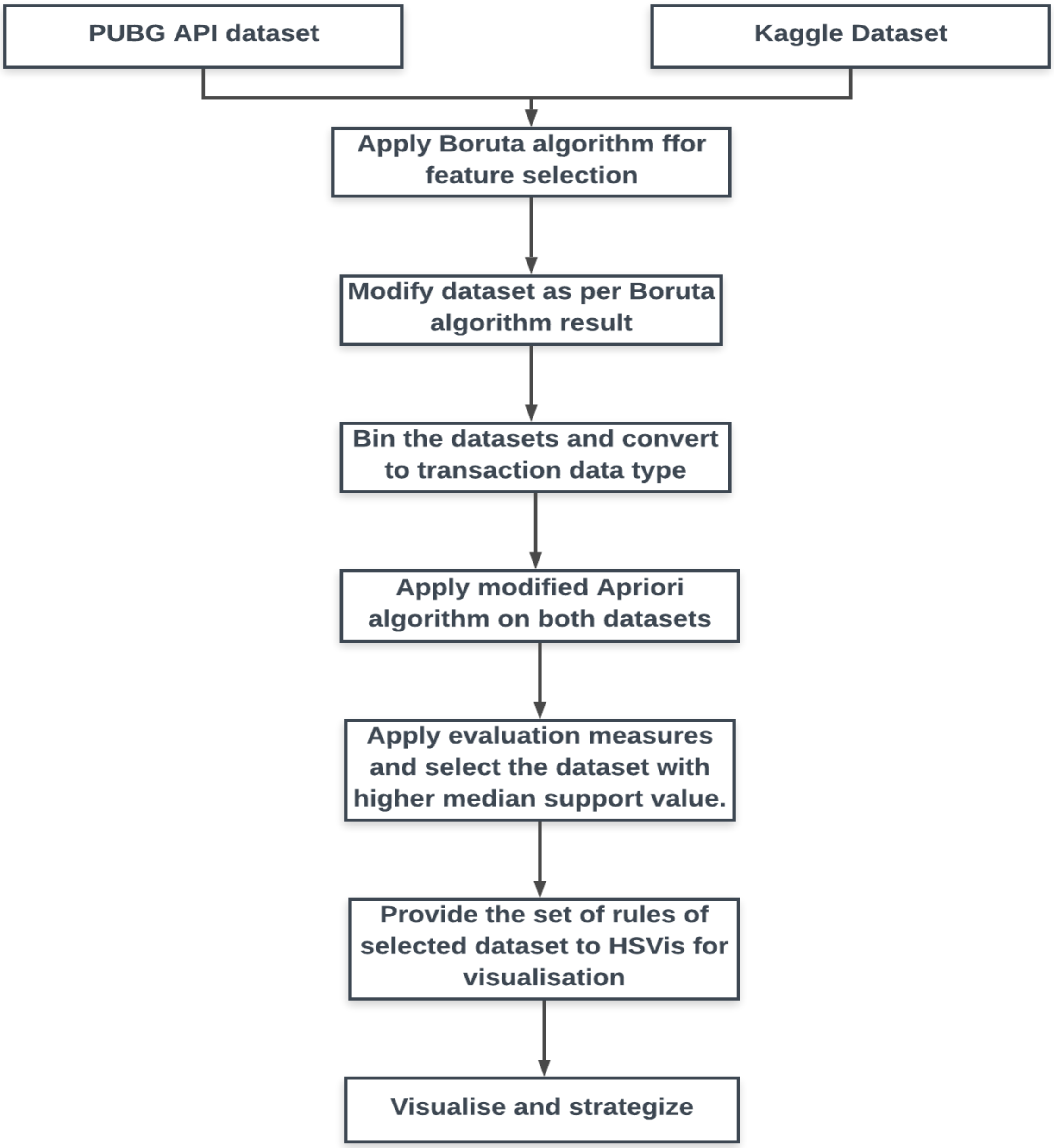


Figure 3: Workflow implantation

5 Implementation

5.1 Data gathering and preparation

To collect data from official PUBG API, a request is sent to get the leader board data which returns players account id and rank. Next a request is sent to get data for all the matches a single player has played. The statistics of a single match of a player is collected and stored in a row. Sequentially all the matches played by a player is added as rows to form one dataframe for one player. These steps are repeated for all the players in the leader board. The data was collected for 400 players and stored as a list.

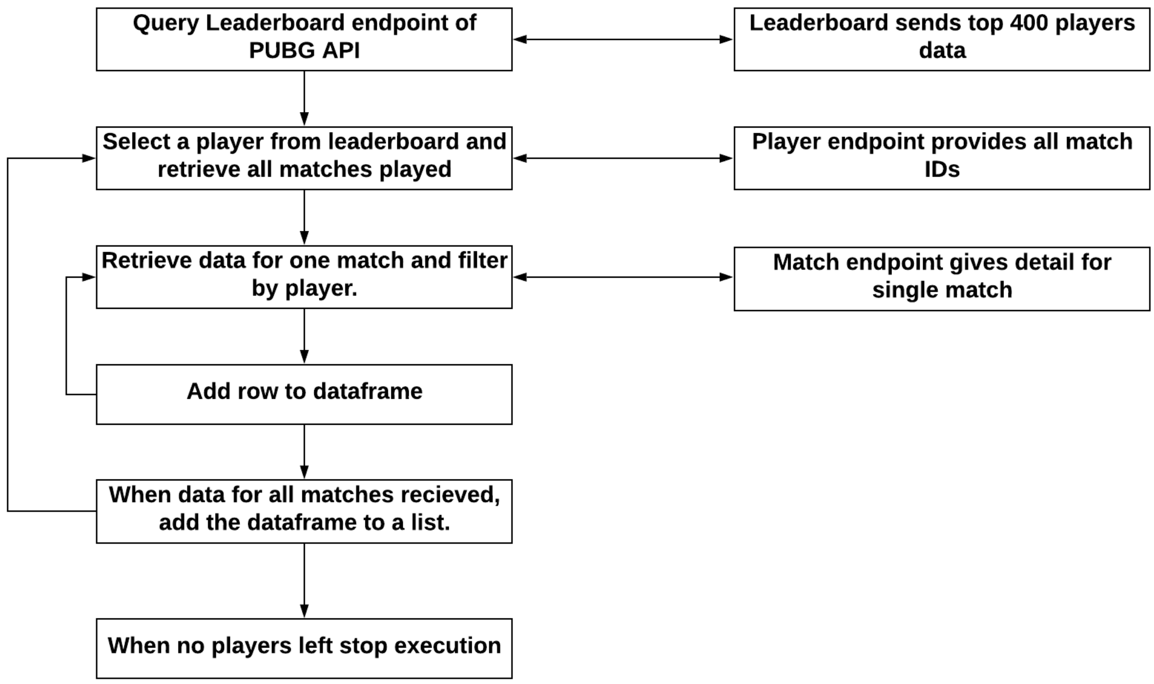


Figure 4: Fetching player details from PUBG API.

List of dataframes is flattened to form one single dataframe containing all statistics of the matches played for all the players.
Kaggle dataset containing player statistics for PUBG, published in October 2018 is divided into train and test data sets. Both data sets are combined to form a single dataset and loaded into the environment.
Initially there are different numbers of attributes in both the datasets. The missing columns in both the dataset are either replaced or deprecated. So the unique column attributes DeathType, MatchDuration, TimeSurvived are discarded. In-game nick and account id attributes are also discarded as they do not aid in further process. The columns are reordered and the datasets are ready for further analysis.

5.2 Boruta Algorithm

Initially Boruta algorithm is applied to the dataset for selecting important features which contribute to the attribute 'WinPlace' which describes the position that player obtained by playing the match. Ideal value for this attribute column is 1. The results indicate that out of

18 variables which may affect the outcome of the match, only one attribute, 'RoadKill' is classified as not important. Thus, the column is discarded. Other 17 attributes are classified as important. No attribute is classified as tentative.

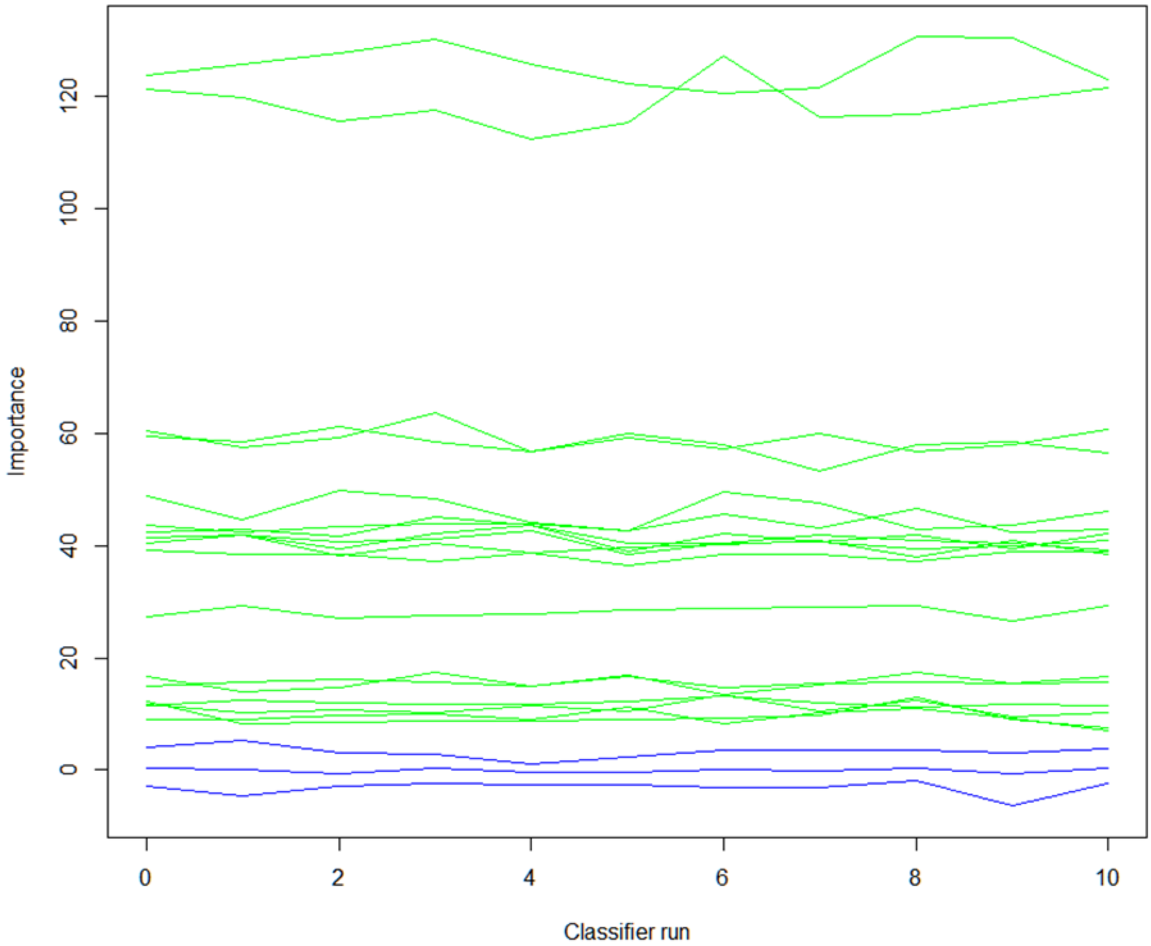


Figure 5: Important attribute history run Boruta algorithm

5.3 Modified Apriori algorithm

Apriori algorithm produces large set of frequently occurring pattern for both the datasets.
To obtain sufficient number of rules, following parameters are experimented with.
Experiments:

Parameters	**Support**	**Confidence**	Number of rules generated
1	30%	30%	> 1 Million rules
2	50%	50%	> 100,000 rules
3	70%	70%	3,281 rules

Table 4: Experimental runs Apriori

The below graphs show the visualisation of the rules which were generated at Support and Confidence values at 70%.

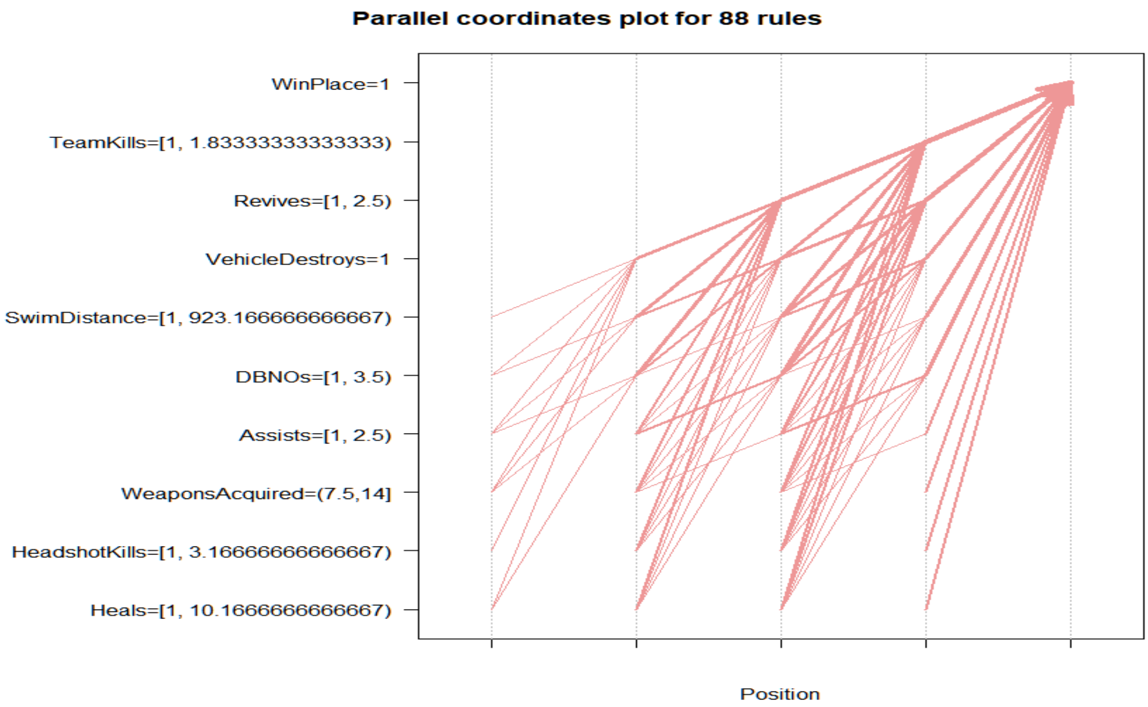


Figure 6: Rules generated for PUBG API.

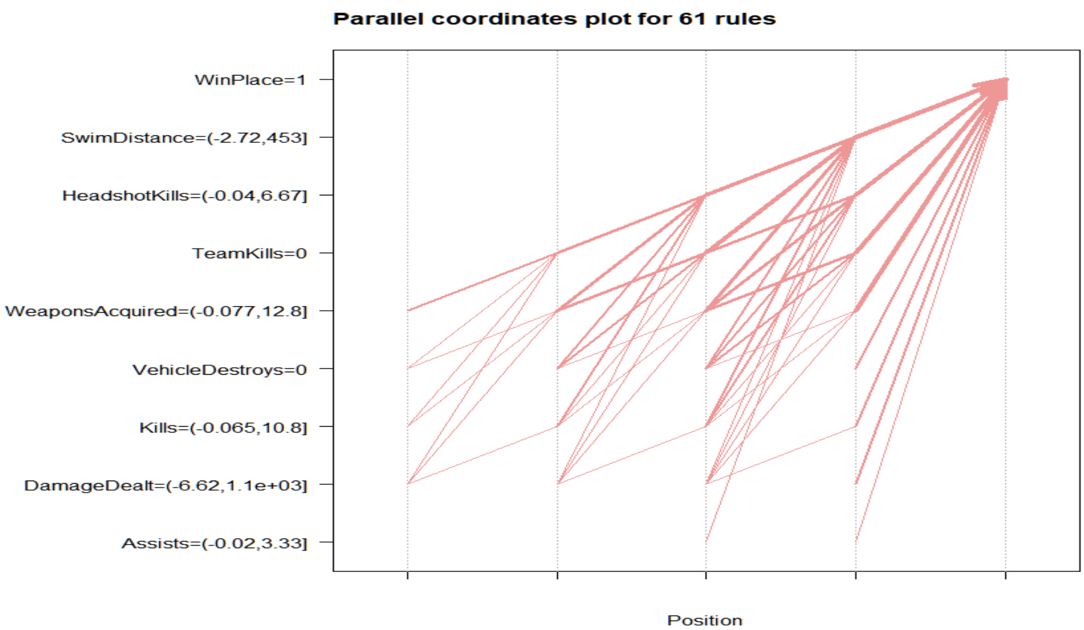


Figure 7: Rules generated for Kaggle Dataset

5.4 HSVis Visualisation

The interactive visualisation built for the frequently occurring patterns.

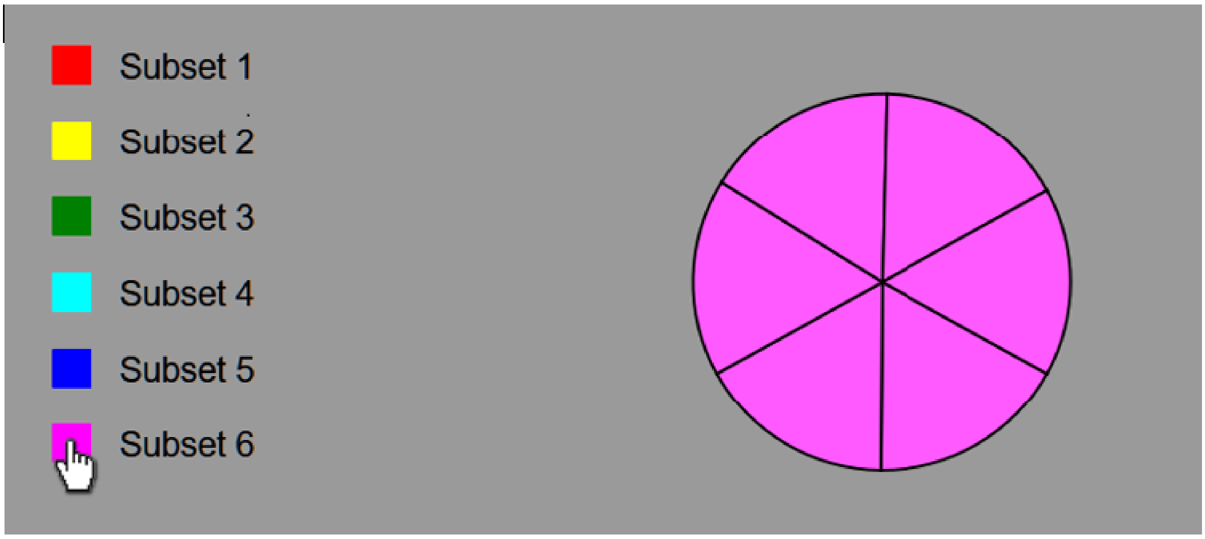


Figure 8: Legend on click interactive visualisation

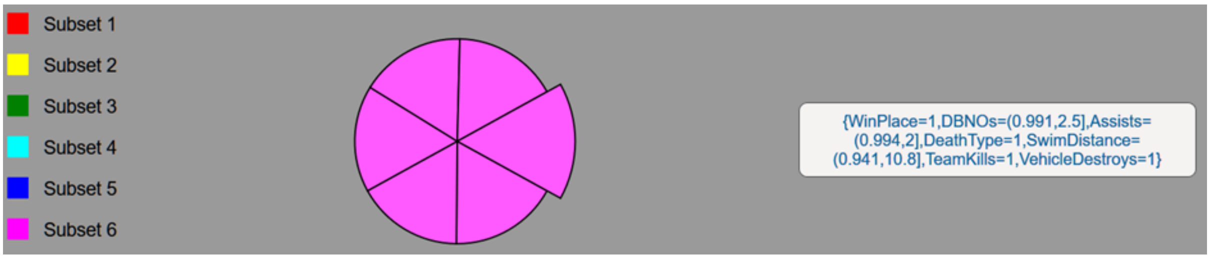


Figure 9: On hover expand and tooltip.

6 Evaluation

6.1 Apriori algorithm execution time

Below statistics provide the minimum time, mean time and maximum time to generate a set of frequently occurring patterns. The unit of time is nanoseconds.

```
apriori(Kaggledf4, parameter = list(support = 0.7, confidence = 0.7,      maxlen = 20, target = "rules"))


expr min     lq     mean     median     uq      max      neval
323 .4972 404.2825 476.0398 454.6965  506.2851 950.0577   100
```

Figure 10: Apriori algorithm execution time in nanoseconds

6.2 Evaluation of frequent pattern generated rules

As seen in the past papers, common methods to evaluate frequently occurring patterns is to calculate support values for the patterns generated by the Apriori algorithm. The support values for both the datasets can be seen in the below patterns.

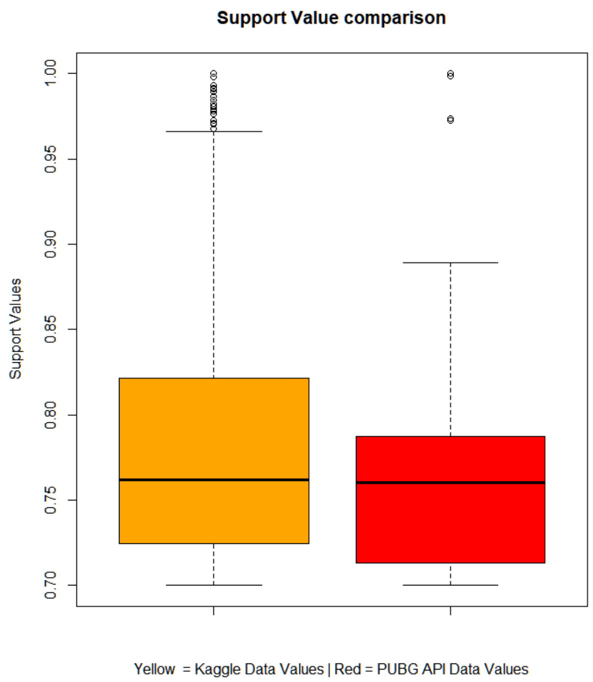


Figure 11: Support Value Comparison

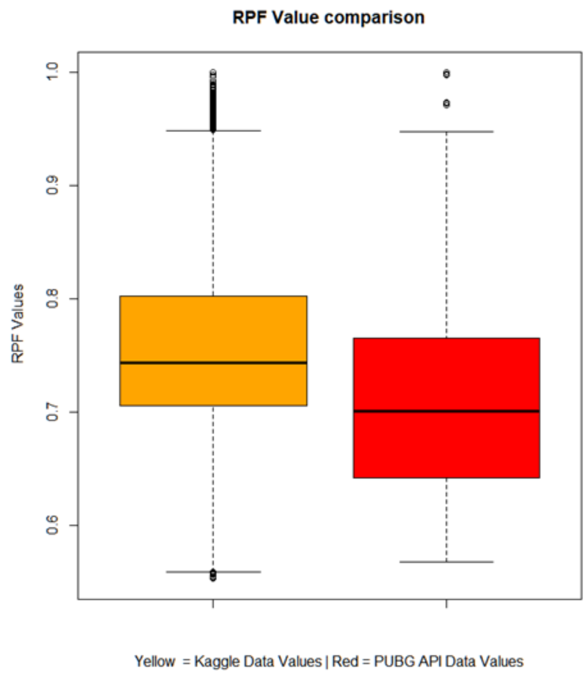


Figure 12: RPF value comparison

On the left, the yellow coloured boxplot represents the support value distribution for Kaggle dataset, whereas, the red coloured boxplot represents the support value distribution for data set created using PUBG API.
The black lines represent the median support values.

In 2016, another measure of interest is proposed, namely Rule Power Factor (RPF) which also evaluates the frequent patterns. The comparison of RPF, is illustrated in the diagram.

RPF correctly classifies less number of interesting rules in PUBG API dataset which were wrongly classified on the basis of support values alone.

6.3 Visualisation load time

The visualisation is built using JavaScript library D3. The time unit for the visualisation is milliseconds.

Page load time for the first instance is 612 milliseconds
Subsequent load time is an average of 513 milliseconds.

7 Discussion

During the evaluation phase the Support measure is calculated by default and the Rule Power Factor (RPF) measure is applied to the generated rules list to check 'interest' factor. These measures indicate the statistical significance of the rules. The RPF measure as described in the 2016 study, makes up for the support and confidence values.

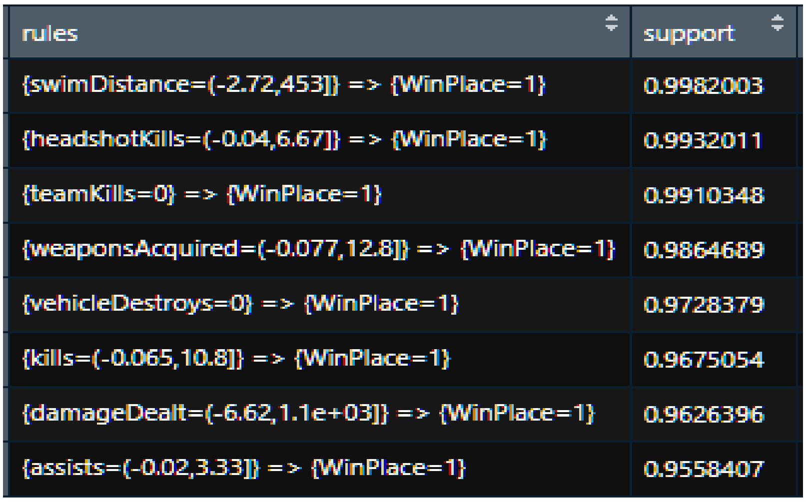
In both the tests, with different measures, we can see that the Kaggle dataset has higher statistical significance in term of support and RPF values when compared with the data set created using PUBG API. Hence, the decision is taken to move further with only Kaggle dataset for the purpose of visualisation and creating a game winning strategy.

One of the reasons, that modified Apriori performed better on Kaggle dataset is because of the size of the dataset. The PUBG API dataset contains approximately 2,100 rows after subset of 84,000 rows whereas Kaggle dataset contains over 30,000 rows after subset of 1 Million rows. We can conclude that the Apriori works better on bigger datasets.

To find the attributes that affect the winning outcome the most, the rules which are generated using modified Apriori algorithm are taken as a subset in which the RHS is 'WinPlace = 1', which indicates that the player wins the particular match.

The rules which indicates the game winning strategy are visualised below.

Next step is to find individual attributes which have a very high probability of affecting the match outcome. The attributes can be seen below image.



rules	support
{swimDistance=(-2.72,453]} => {WinPlace=1}	0.9982003
{headshotKills=(-0.04,6.67]} => {WinPlace=1}	0.9932011
{teamKills=0} => {WinPlace=1}	0.9910348
{weaponsAcquired=(-0.077,12.8]} => {WinPlace=1}	0.9864689
{vehicleDestroys=0} => {WinPlace=1}	0.9728379
{kills=(-0.065,10.8]} => {WinPlace=1}	0.9675054
{damageDealt=(-6.62,1.1e+03]} => {WinPlace=1}	0.9626396
{assists=(-0.02,3.33]} => {WinPlace=1}	0.9558407

Figure 13: Frequent pattern rules of cardinality 1.

From above, we can conclude

- SwimDistance (2 meters to 453 meters) - If there is water body nearby, travelling through it should be given a preference rather than on foot or using a vehicle.
- headshotKills (1 to 7 kills) – High accuracy is a big part of the game. If there is an option to select different guns, the one which delivers headshots should be preferred. Ideally kills between 1 to 7 ensures winning.

- TeamKills (killing a team mate) – Not reducing the health points or killing a teammate is essential to winning.
- WeaponsAcquired (1 to 13) – Having more than one weapon is essential.
- Kills (1 to 11 enemies) – It is expected to face 11 enemies or more. Not all of them can be eliminated via headshots, so a secondary gun is required.
- DamageDealt (7 to 1103 health points) – Dealing maximum damage without killing enemies is also sufficient for elimination.
- Assists (0 to 3) – Since we face high number of enemies, damage dealt to enemies will help teammates eliminate them.

Above rules are statistically proven to be significant derived from the given dataset, as seen in the support values.
The custom visualisation which is developed enables to interact with the patterns containing up to six attributes in a single set.

8 Conclusion and Future Work

This research study has proposed a strategy which will enable players to win the game, laid a foundation for other similar games which can visualise winning patterns, developers can analyse and offer players in-app purchases suited to their playing style based on the visualisation and made improvements to the existing visualisation making it interactive and easy to access and understand.

One of the limitations of this visualisation is that since there are similar looking visualisation, and the user maybe confused about its usage. Therefore, more information about working of the visualisation needs to be provided to the user through an onscreen tutorial or a manual which adds to the overhead. Although the visualisation can show up to six frequent items in a single set, there is possibility that itemset with cardinality greater than six maybe present, having higher support value, which should be given a preference. For future work, researchers can build on this to highlight such anomalies and give preference to the higher support values in visualisation.

Another aspect which can make games more immersive is real time guidance to the players. If a player strays away from the team, warning sign and an arrow leading to the team's location will guide the player, weapons with better accuracy should be highlighted or nearest water source should be highlighted This methodology can be extended to other games like Tom Clancy's rainbow siege or Modern Warfare 2.

Acknowledgements

I would like to thank Prof. Vladimir Milosavljevic for his guidance in implementing of this research study and helping me look at the bigger picture which enabled me to contribute to the field of data analytics.

References

Babi, C. (2018) ‘Mining Frequent Patterns from Big Data Sets using Genetic Algorithm’, 11(2), pp. 287–306.
Babi, C. *et al.* (2018) ‘Using visual analytics to detect problems in datasets collected from photo-sharing services’, *Procedia Computer Science*. IEEE, 126(44), pp. 720–726. doi: 10.1109/IAdCC.2014.6779361.
Barkwell, K. E. *et al.* (2018) ‘Big data visualisation and visual analytics for music data mining’, *Information Visualisation - Biomedical Visualization, Visualisation on Built and Rural Environments and Geometric Modelling and Imaging, IV 2018*, pp. 235–240. doi: 10.1109/iV.2018.00048.
Bikakis, N. and Sellis, T. (2016) ‘Exploration and visualization in the web of big linked data: A survey of the state of the art’, *CEUR Workshop Proceedings*, 1558.
Borgelt, C. and Kruse, R. (2011) ‘Induction of Association Rules: Apriori Implementation’, *Compstat*, pp. 395–400. doi: 10.1007/978-3-642-57489-4_59.
Braun, P. *et al.* (2017) ‘Game Data Mining: Clustering and Visualization of Online Game Data in Cyber-Physical Worlds’, *Procedia Computer Science*. Elsevier B.V., 112, pp. 2259–2268. doi: 10.1016/j.procs.2017.08.141.
Brin, S. *et al.* (2005) ‘Dynamic itemset counting and implication rules for market basket data’, *ACM SIGMOD Record*, 26(2), pp. 255–264. doi: 10.1145/253262.253325.
Dharani, S. *et al.* (2016) ‘Performance comparison of frequent pattern mining algorithms for business intelligence analytics’, *Indian Journal of Science and Technology*, 9(44). doi: 10.17485/ijst/2016/v9i44/98864.
G, S. M. *et al.* (2016) ‘PRE-PRINT:CALI: A novel visual model for frequent pattern mining in protein-ligand graphs’, pp. 1–8. doi: 10.1109/BIBE.2017.00066.
Heaton, J. (2016) ‘Comparing dataset characteristics that favor the Apriori, Eclat or FP-Growth frequent itemset mining algorithms’, *Conference Proceedings - IEEE SOUTHEASTCON*, 2016-July. doi: 10.1109/SECON.2016.7506659.
Jamsheela, O. and Raju, G. (2015) ‘Frequent itemset mining algorithms: A literature survey’, *Souvenir of the 2015 IEEE International Advance Computing Conference, IACC 2015*. IEEE, pp. 1099–1104. doi: 10.1109/IADCC.2015.7154874.
Jayaprada, S. *et al.* (2014) ‘VizSFP: A visualizer for semantically similar frequent patterns in Dynamic Datasets’, *Souvenir of the 2014 IEEE International Advance Computing Conference, IACC 2014*. IEEE, pp. 422–427. doi: 10.1109/IAdCC.2014.6779361.
Leung, C. K. S., Jiang, F. and Irani, P. P. (2011) ‘FpMapViz: A space-filling visualization for frequent patterns’, *Proceedings - IEEE International Conference on Data Mining, ICDM*. IEEE, pp. 804–811. doi: 10.1109/ICDMW.2011.86.
Ochin, Kumar, S. and Joshi, N. (2016) ‘Rule Power Factor: A New Interest Measure in Associative Classification’, *Procedia Computer Science*. The Author(s), 93(September), pp. 12–18. doi: 10.1016/j.procs.2016.07.175.
Siqueira, E. S. *et al.* (2018) ‘A data analysis of player in world of warcraft using game data mining’, *Brazilian Symposium on Games and Digital Entertainment, SBGAMES*. IEEE, 2017-November, pp. 1–9. doi: 10.1109/SBGames.2017.00009.
Leung, C., Irani, P. and Carmichael, C. (2008) ‘FisViz: A Frequent Itemset Visualizer. *Advances in Knowledge Discovery and Data Mining’*, pp.644-652.
Leung, C., Kononov, V., Pazdor, A. and Jiang, F. (2016). PyramidViz: Visual Analytics and Big Data Visualization for Frequent Patterns. *2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing,*

2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech).
Leung, C. and Carmichael, C. (2010). FpVAT. *ACM SIGKDD Explorations Newsletter*, 11(2), p.39.
Carmichael, C. and Leung, C. (2010). CloseViz. *Proceedings of the ACM SIGKDD Workshop on Useful Patterns - UP '10*.
Leung, C., Irani, P. and Carmichael, C. (2008). WiFIsViz: Effective Visualization of Frequent Itemsets. *2008 Eighth IEEE International Conference on Data Mining*.
Li Min Fu and Shortliffe, E. (2000). The application of certainty factors to neural computing for rule discovery. *IEEE Transactions on Neural Networks*, 11(3), pp.647-657.