Jenny Reschke

Final Report:
Catching Hackers in Call of Duty: Modern Warfare

Problem Statement

The popularity of Call of Duty (CoD) games stems from the ability to compete against other players from around the globe with equal footing. In fact, *Call of Duty: Modern Warfare 2* is a multiplayer video that sold over $1 billion in sell-through worldwide in the first 50 days of the game's arrival **(Activision Blizzard)**. In the era of COVID-19 where social distancing is encouraged, CoD has become even more of a staple for people to socialize with friends and pass the time while stuck in their homes.

Call of Duty: Modern Warfare 2 consists of teams of players ranging from 2 to hundred competing against another team in various war-like competitions. Some of the goals vary based on the type of match, but one of the ultimate metrics of a good player is their kills to deaths ratio (KD). A high KD means the player is skilled enough to kill more players in a match than the number of times the player died in the match. Though the game has a multitude of customizations like weapons and special abilities, the determining factor should be the player's skill at the game. With its popularity, CoD hopes to continue to expand and an established esport.

But, *Call of Duty: Modern Warfare 2* suffers from a significant hacking problem where players can mess with code of the game and have impossibly high KDs (ScreenRant). This affects the enjoyment of playing the game as all other players will continue to die over and over again without any ability to stop the hacker. Activision, makers of CoD, are aware of the problem and have blacklisted hacker's accounts from their servers, but the prevalence of hackers continues to be a problem.

Using data gathered from CoD API and scraping player ranking data from CoD Stats Tracker, this project will look at the abnormalities in the players statistics to determine if the player is a hacker or not. These abnormalities will be considered extreme outliers that are not practical even for the professional player. For example, a player with a perfect or near-perfect accuracy. Though this project focuses on hacking in a video game, the idea of identifying extreme outliers in user behavior could be applied to other hacking activity like credit card fraud.

This project utilizes a variety of supervised and unsupervised techniques including Local Outlier Factor (LOF), KMeans Classification, and RainForest Classification. This resulted in PRECISION for predicting outliers in the data set that could be applied to future hacker detection efforts in CoD and other video games.

Data Wrangling

The data set was produced from scraping gamertags (usernames in CoD) from CoD Stats Tracker top players list. The list is based on the website's own ranking system that combines the rankings of each player for their individual game statistics like KD, accuracy, games won, etc. The scrapper took the top-120 gamertags and compiled them into a CSV.

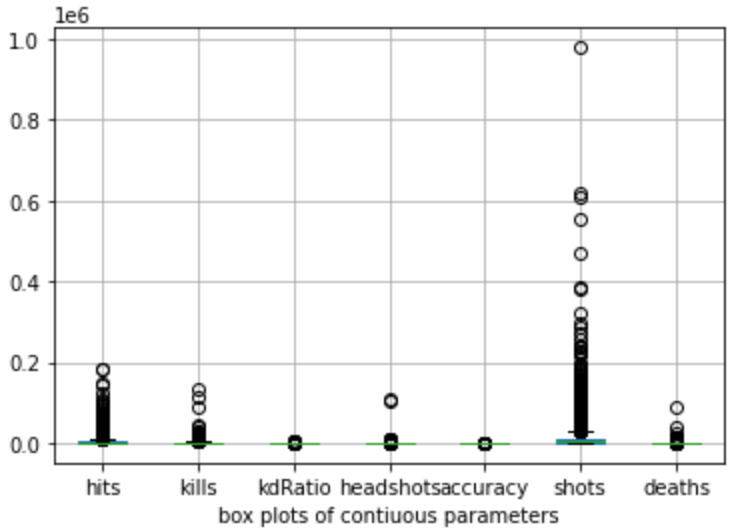
To get the player's statistics, the CSV was fed into an API that produced a data set containing all-time statistics for each weapon in the game for each player on the list. The raw data set consisted of 4,732 rows of 12 columns. The columns consisted of the in-game statistics

like hits, kills, KD ratio, headshots, accuracy, shots, and deaths and other categorical parameters such as the weapon, the type of weapon, the gamertag and type of console the players used.

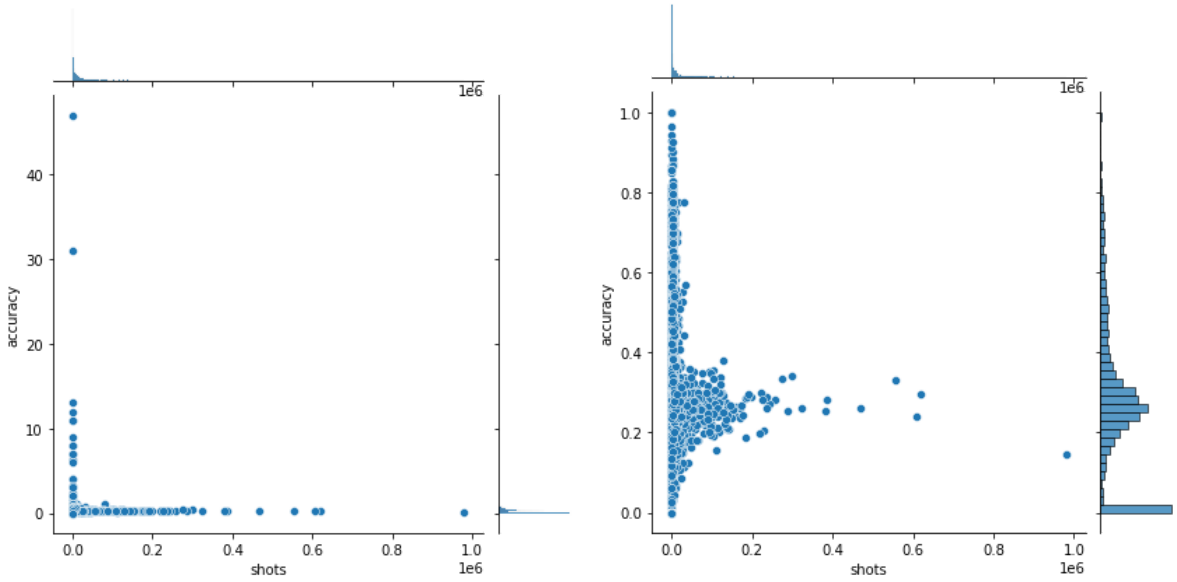
One of the columns was over 90 percent empty so it was removed from the data set. Some of the weapons like melee weapons do not record all of the statistics like the accuracy metrics because it will always hit if utilized in the game. Since melee weapon type is different from other gun-based weapons, it was excluded from the data set. This reduced the data set to 3,676 rows and 11 columns.

Exploratory Data Analysis

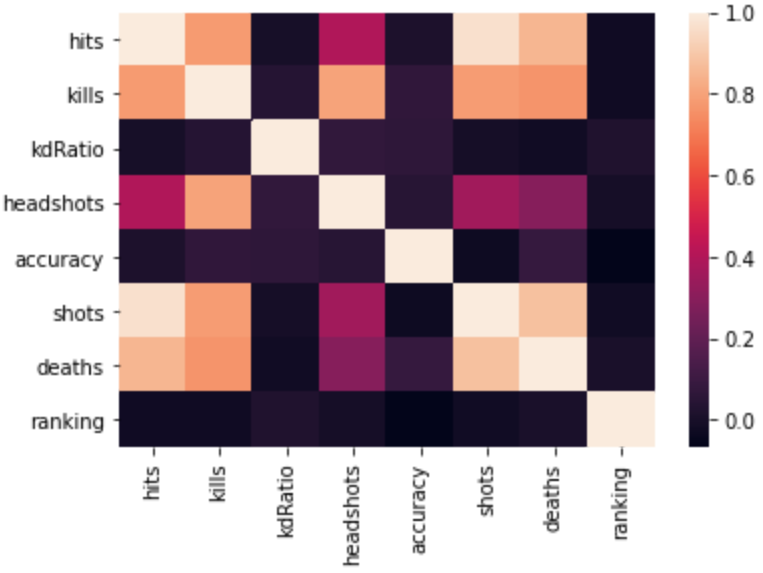
To start exploring the data set, let's explore the distribution of the input parameters. As shown in the box plot below, there are some significant outliers for each input parameter, especially the number of shots parameter. The substantial number of outliers for shots is not abnormal as it is the cumulative total of all the shots taken by a player since the game came out. If a player is trigger-happy, then they could have an extreme large number of shots taken.



To verify that the extreme shot values are not outliers that should be removed from the data set, the next plot shows the total number of shots against accuracy. Assuming the player has a high number of shots taken, then their accuracy should be relatively poor. The plot belows shows this to be the case. But looking at the y-axis, it appears that there are values of accuracy that go above 1 which is infeasible. Since these values are not possible, they are removed from the data set. The next plot shows the result after removing the accuracy outlier values. The right shows a normal looking distribution for accuracy which matches the expectations of that parameter.



Finally, a heat map is created to see what possible correlations exist between parameters. This information could help in determining the best model to use during analysis. Ignoring the categorical variables such as weapon type, it appears only hits and shots have a significant correlation. Parameters that measure the player's skill like KD ratio, accuracy or ranking appear to have little or no correlation to the other parameters. A possible reason for this could be that this data set does not fully capture all that is involved in a player's skill, but this is a topic for future research.



Feature Selection

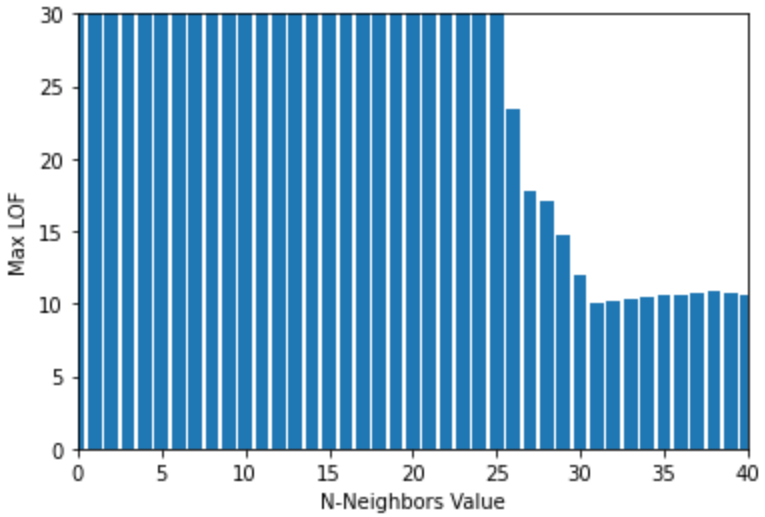
To prepare the data set for analysis, a number of feature selection tasks were performed. All the continuous input parameters were scaled to a 0 to 1 range. The categorical variables are replaced by dummy variables. This means each level of a categorical variable gets

its own column where a *one* value means the row is at the given level and a *zero* means the row is not at the given level. Dummy variables make categorical variables compatible with classification models. Finally, the data set was divided into a training (2730 rows) and testing (911 rows) data set.

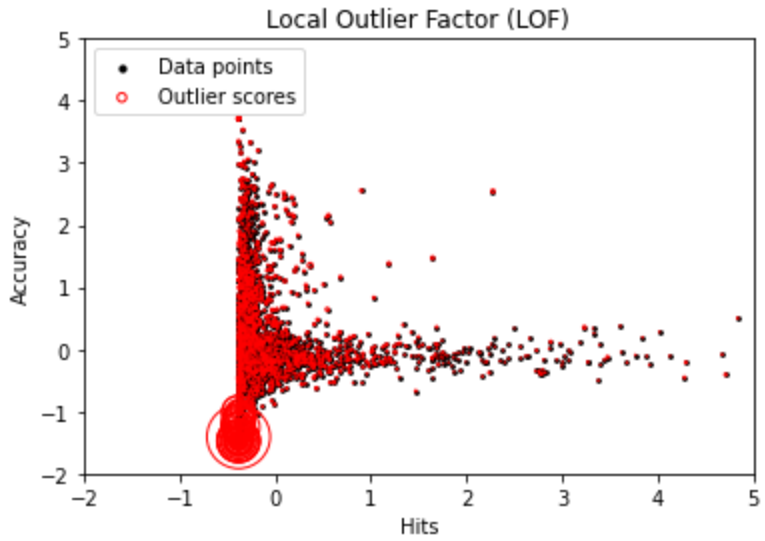
Model Selection

Since this is a raw data set, the first step is determining which points are the outliers. To do this, an unsupervised anomaly detection method is applied to label if a point is not an outlier or not. Unsupervised learning means no prior knowledge or labeling is included in the model development. This project used the local outlier factor (LOF) to determine the outliers for the classification.

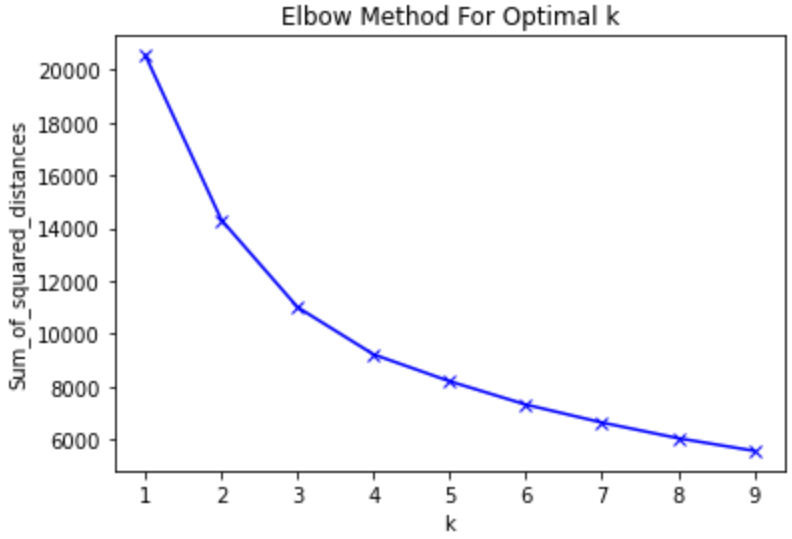
LOF determines if a point is an outlier based on how close the point is relative to other points in the sample space. The distance between one point and all of the neighboring points determines its outlier score. The number of neighboring points that are considered a cluster is the one hyperparameter that can be tuned in a LOF. To determine the ideal number of neighbors, the range of 10 through 1,000 was run through the LOF with the training data set, and the maximum value of the outlier factor was saved for each iteration and plotted. The plot below shows results of the number of neighbors against the maximum outlier factor value. The goal is to find the elbow of the plot and that value will be used as the hyperparameter. Based on the plot below, 31 neighboring points is the ideal for cluster size.



With this determination, LOF is run on the training data set and applied to the test data set. If a point has an outlier factor value greater than 1.5, then it is determined to be an outlier. The outlier or not column is added to both data sets as output parameter for both the KMeans and Random Forest classifiers. The plot below shows the training data set scatter plot where the red circle represents the outlier factor value where a larger circle means a larger outlier score.



This project uses both a KMeans and a Random Forest classifier as possible models. KMeans classification model is similar to a LOF model as it depends on the location of the point to its neighboring points. Instead of determining the number of neighboring points, KMeans's one hyperparameter is the number of expected clusters in the data set. To determine the ideal value of k, a number of k values are run through the KMeans with the training data set. The sum of squared distance between points in a cluster is calculated and the results are plotted. The kink or elbow in the plot is the optimal value of k which in this case is 3. With the KMeans hyperparameter tuned, the model can be run with k equal to 3, and predictions can be made with the test data set. Table 1 shows the results of the Kmeans = 3 model.



To improve upon the KMeans classifissier, a bagging method is employed. Bagging takes a subsample of the training data set and builds a model. It does this multiple times as each subset could capture a different aspect. The final result is combined by averaging all the models together. Based on the results in Table 1, KMeans with the bagging method improved the overall precision and accuracy of the model.

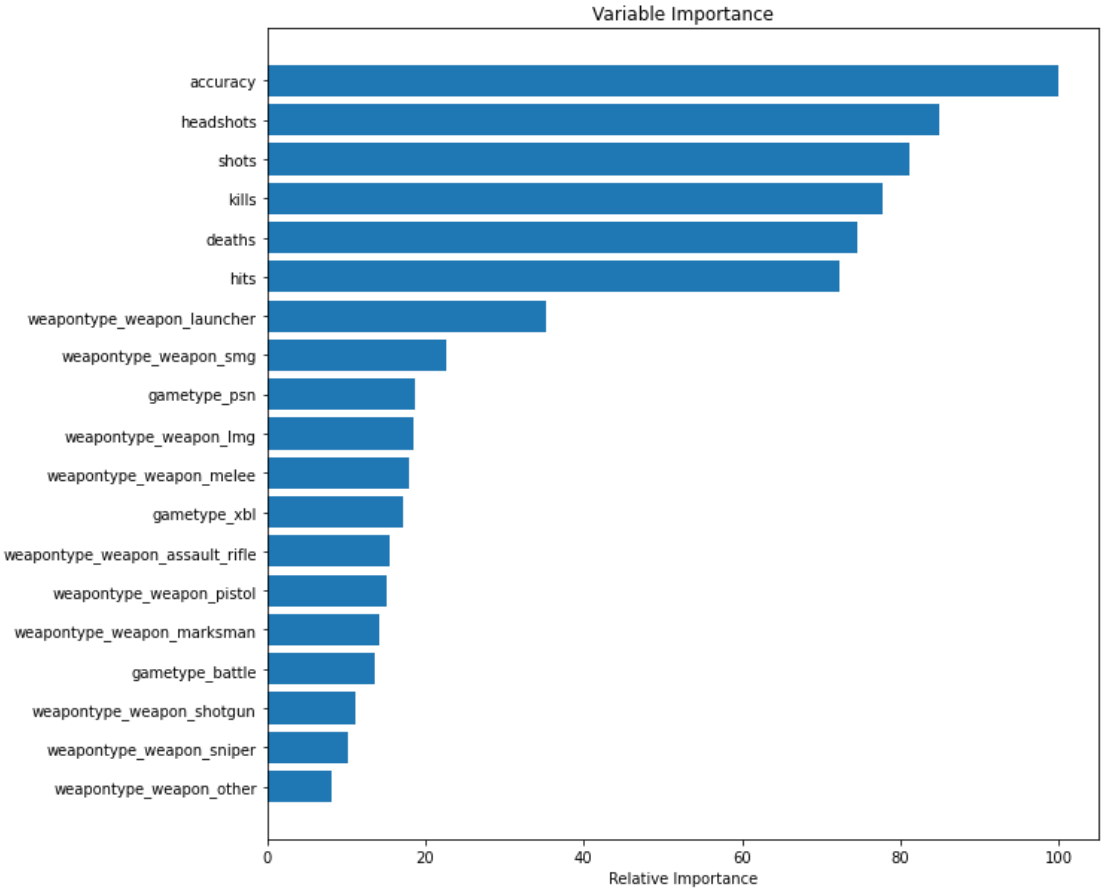
Another classification method is using a random forest. A random forest is made up of multiple decision tree models. These decision trees take a random sample of the training data set, much like the bagging method, but they also select a random number of input parameters. Then the decision trees are averaged resulting in the final result of a random forest. By performing a grid search, the optimal number of decision trees was determined to be 155. Based on Table 1, the random forest classification performed the best in terms of the recall for the outlier classification and the KMeans with bagging classification model performs best for the precision score of the outlier points. Since the goal is to correctly identify hackers and then remove them from the game, the accuracy metric should focus on correcting identifying outliers, meaning a false positive is more dangerous than a false negative. Therefore, the KMeans with Bagging classification model's precision score means it's the more useful model.

Table 1

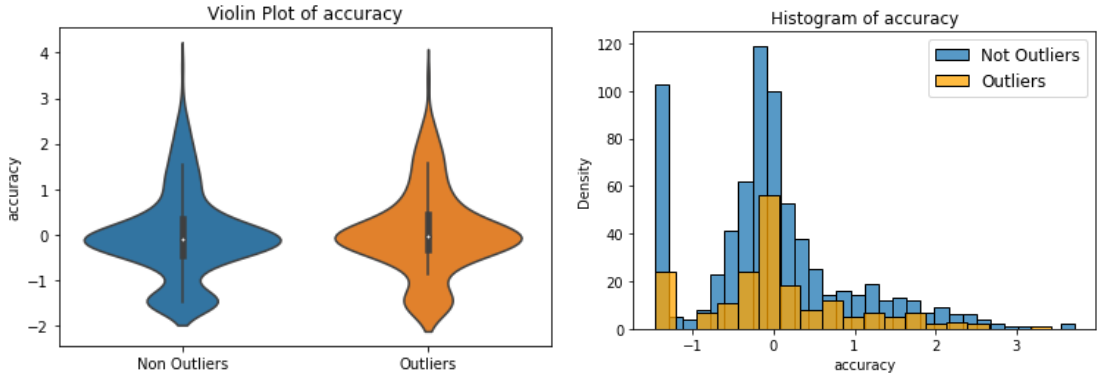
	KMeans	**KMeans + Bagging**	**Random Forest**
Accuracy	0.75741	0.78704	0.70032
Balanced Accuracy	0.47983	0.50433	0.49329
Precision score for non outlier points	0.79401	0.79069	0.78663
Precision score for outlier points		0.37500	0.19548
Recall score for not outlier points	0.95966	0.99304	0.85118
Recall score for outlier points		0.01562	0.13541

Takeaways

Beyond making predictions, the random forest model contains information on the importance of each input parameter to determining whether the point is an outlier or not. Based on the results shown in the plot below, accuracy is the most important parameter followed by headshots, shots, and kills. As shown in the exploratory data analysis, the accuracy input parameter had a substantial number of outliers where the accuracy value was over 1.0. Furthermore, it makes sense that whatever the hackers do would improve their accuracy as it would help them win the match. The variable importance plot also shows that the type of weapons or the type of device the player uses does not have a significant effect on the outliers.

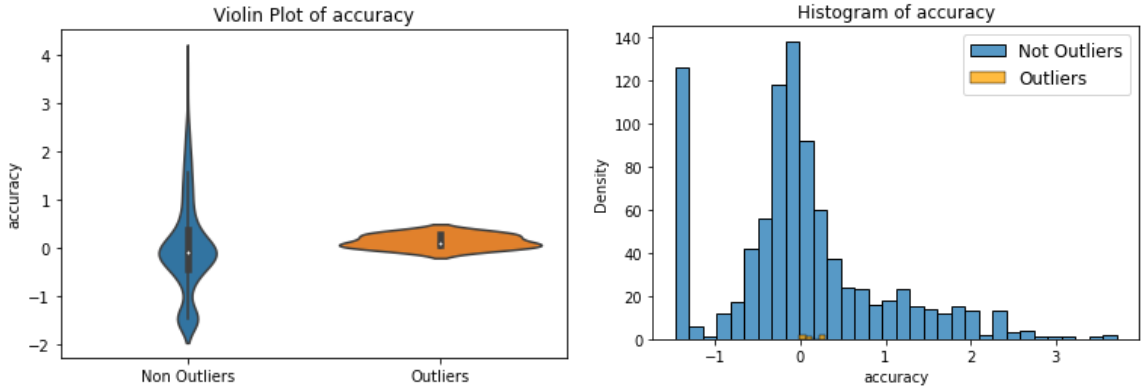


Looking at the distribution plots with the LOF classification, it appears that the outliers and not outlier points follow the same general distribution for the accuracy input parameter. This is the same for all of the continuous input parameters, as there is no clear pattern or section of the sample space that the outliers reside. This may explain the low accuracy metrics for all the models.



Looking at the predictions for KMeans with bagging model, the results show a clear section of the sample space where the outliers occur. For example, it appears that the predicted outliers have a normalized accuracy value around 0, (see the plots below). Since the same plots vary significantly, this could explain why the fitting metrics are poor. In future variations of this

project, other ways of performing unsupervised classification should be explored to see if they can better identify outliers and in turn produce better fitting classification models.



Future Research

This project performed some preliminary analysis on determining outliers and therefore, the hackers in *Call of Duty: Modern Warfare*. The current models provide some initial insights into the behavior of the outliers but there is always room for improvement and further research.

Future studies could start with the initial data set but explore the possibility of other input parameters that involve the skill of the player. As based on this project, the statistics like accuracy and headshots played a greater role in identifying outliers than the other input parameters. The next interaction could include parameters like the win to loss ratio, kill to death ratio, or damage taken compared to damage delivered. These are all skills that separate good from bad players but also can show players that may have unbelievable and therefore hacked statistics.

Another future area of study would be exploring other methods of unsupervised learning to initially label the outliers in the data set. Furthermore, if available, it would be good to compare against an existing hacker list to see if the unsupervised algorithm is able to correctly identify outliers in the data set.