December 6, 2022

Jeff Sit

Seneca College

BDM500

Predictive Analytics

final project report

BDM500 Project MS1

**Objective**:   
To analyze at the highest levels of play, the settings that professional Overwatch esports players compete with. While there exist many different settings like control schemes, peripheral layout, and monitor setup, the main focus of this project will be on mouse settings. Specifically, this includes but is not limited to, the model and weight of the player’s mouse, the player’s mouse sensitivity measured in Dots per Inch (better known as DPI), and the player’s in-game sensitivity usually represented as a decimal number.

**Intended Outcomes:**The settings of 148 professional players will be analyzed to determine what the common (and thus the most competitive) mouse settings are for some of the best players in the game. Specific examples of what can be determined would be if there is a correlation between a player’s in-game role, the distance it would take them to perform a 360 degree turn (cmPer360), and the mouse settings they use.

It would be plausible to predict that a player filling the role of a fast flanker will need a higher DPI and in-game sensitivity so that they are able to keep their mouse movements in sync with their hero’s speed. In contrast, it would also be reasonable to assume that players on the opposite end of that role, say they were playing a much slower but tougher hero, would have a slower, more manageable DPI and sensitivity as their mouse movements would not be focused on aiming over targets in quick succession. For players that use heavier mice, it may be possible that they will typically use higher mouse speed settings to compensate for the heavier weight of the mouse. Essentially, it will be determined whether or not a player’s in-game role can be predicted based on other attributes present in this dataset.

**Prior Knowledge Required of Audience:**A basic understanding of how multiplayer hero-based games work, primarily in how there are three roles (classes) that make up a team in this particular game and that there are many heroes within each role, each of which play differently. Surface level knowledge of how mouse devices work is also needed, understanding that the higher the DPI or in-game sensitivity, the faster the mouse pointer on the screen will move. In the case of a game using a first-person camera (which this game does), then higher DPI/sensitivity will result in faster and usually harder to control camera movements.

**Foreseeable Challenges:**Some challenges would include how some players may prioritize comfort over all else, including what is widely considered to be good or optimal for their role or mouse. Therefore, there may be contradictions where some support players may have a higher overall sensitivity while some offensive players may have slower overall sensitivities. This discrepancy might even extend to include mouse weights as well, where some pros prefer older mice that are technologically inferior to modern mice. The only reason for their usage is due to the unique shape provided by the older mouse, and potentially years of familiarity and mastery attained on that mouse over time. Therefore, should these issues be significant enough, it is possible there might not be enough data to go off of as only 6 numerical variables will be used in this dataset for prediction purposes.

BDM500 Project MS2

**Dataset Description:**This dataset contains information on 148 professional Overwatch players’ main role and hero they play, as well as the mouse they use and what settings they personally opt for that affect mouse settings. A list of categorical and numerical variable names used can be found below.

List of categorical variables (4): Player, Role, Hero, MouseModel.

List of numerical variables (6): MouseWeight, MouseWeightToSizeRatio, DPI, Sensitivity, eDPI, cmPer360.

Pictured below is a sample visualization created early on in Tableau Public. It plots all of the heroes against the average eDPI used by the professional players playing these characters.

Graphical user interface, chart, bar chart

Description automatically generated

**Foreseeable Challenges:**It is possible that this dataset might not contain enough information in the way that there are only 6 numerical variables. Additionally, there are only 148 rows, which when split between training and test data, does not amount to a lot of data for either component. Therefore, it might be difficult to meet the requirement of applying a minimum of 2 different predictive analytics techniques.

Luckily, it appears that is where the doubts about this dataset end. Since this is a standard csv file, there should be no issues with regards to incompatible data or errors in importing the data. Also, with numerous data entries in the form of 148 professional Overwatch players recorded, this is the perfect sample size to perform this type of project on.

BDM500 Project MS3

**Exploratory Analysis:**To perform this step, basic stats about the dataframe are retrieved using commands like .shape, .info(), .describe(), and .value\_counts().

**Identifying metrics in the data and creating KPIs:**Some important metrics that will be used include the number of players using the same model of a particular mouse, the average mouse weight professional players gravitate to, and of course, the cmPer360 value that professional players generally stay around.

The primary goal of this dataset is to determine if there is a relationship between a player's mouse distance required to perform a 360 degree turn and their in-game role. The key performance indicators that will be used to gauge the success of this dataset will combine at least 2 metrics. As an example, the player's cmPer360 value will be taken and compared to the role they play. To reiterate, these KPIs aim to determine the relationship of cmPer360 to player Role.

**Cleaning, filtering, and editing data:**While there is no data that needs to be filtered, there exist a few records that need to be cleaned/edited. These records have structural errors where records of the same type follow different naming/spelling conventions (e.g FinalMouse vs Finalmouse). This leads to these records being counted separately, ultimately altering the results of the analysis if not dealt with during the initial stages.

**Ensuring data quality and validity:**Simply combining .isnull() and .sum() as .isnull().sum() will show that the data has no missing values or records. Overall, the data meets all criteria for use as it is accurate, relevant, complete, up-to-date, and now follows a consistent format.

**Imputing missing data using linear regression, mean, and ratios:**As there are no missing values or records, imputation through linear regression or mean is not necessary. However, assuming it was, the following commands would be used to resolve this issue.

for categorical data

* from sklearn.impute import SimpleImputer
* imp = SimpleImputer(strategy = "most\_frequent")
* df["Hero"] = (imp.fit\_transform(df["Hero"]))

for numerical data using mean

* mean\_imp = SimpleImputer(strategy = "mean")
* df["DPI"] = (imp.fit\_transform(df["DPI"].values.reshape(-1,1))

for imputation using linear regression

* lr = LinearRegression()
* lr.fit(xtrain, ytrain)
* xtest = test\_data['MouseWeight']
* ypred = lr.predict(xtest)
* test\_data['ypred'] = ypred
* test\_data = df[df["DPI"].isnull()]
* df.dropna(inplace=True)
* xtrain = df.drop('DPI', axis=1)
* ytrain = df['DPI']
* from sklearn.linear\_model import LinearRegression

BDM500 Project MS4

**Calculating skewness of dataset and interpreting the values:**All variables except for MouseWeight and MouseWeightToSizeRatio have a positive skew. This can be explained easily, as the physical characteristics of the mouse these players are using cannot be significantly changed. Considering how every pro player is using a modern gaming mouse such that each are designed by the company to be as light as possible, it should come as no surprise that they all hover around a mostly symmetrical distribution. Mouse size is slightly different but mostly the same in this case, as gaming mice come in different sizes to suit the different hand sizes and grip styles of the user. Again, the mice are designed by the company, so there is no personal input from the player that can affect this value.

DPI, Sensitivity, eDPI, and cmPer360 all have a positive skew. The reason for this skewness is simple, as mice settings have a setting called DPI built into them that stands for Dots Per Inch, which governs how far the pointer should move given one inch of mouse movement. Higher does not mean better however, as while the mouse pointer can move faster, it will also become harder to control and prone to error. This is why there is a positive skew for this variable, as most people will tend to gravitate to DPI values between 400 to 1600. However, there are certainly players that play on much higher DPIs, which explains the tail of this skew being on the right side.

To compensate for these higher DPIs, players will typically opt for a decreased Sensitivity. This is because ultimately, the overall mouse movement per inch of mouse movement is equal to DPI multiplied by Sensitivity. In other words, effective DPI or eDPI = DPI \* Sensitivity. This explains the massive positive skewness of eDPI, as the multiplied values can increase very quickly.

Lastly, cmPer360 is positively skewed as well since there is an inverse relationship between this variable and eDPI. Therefore, as eDPI increases, cmPer360 decreases, as less movement is required to execute each maneuver.

**Constructing and interpreting histograms and plots.**Chart

Description automatically generated

As mentioned before, when it comes to mouse weight, not much can be done about this as everyone is getting the same specifications according to the mouse model chosen. Clearly, most pros prefer mouse weights of around 80 to 90 grams with some players having access to much lighter mice at 60g and others preferring to stick with heavier mice at around 110g. Pro players do not use mice heavier than 120 grams and nothing lighter than 50 grams, the latter of which is likely because there is simply no mouse that light that exists right now. Note that the above visualization is a recreated version of the histogram seen in the jupyter notebook. It appears as a column chart since the option to choose a histogram does not exist in Power BI.

Chart

Description automatically generated with low confidence

For DPI it is very common for pros to pick between 400, 800, or 1600. Numbers in between are typically never chosen. This is because historically, gaming mice never had customizable software that players could use to change these values. As a result, older gaming mice that these pro players likely grew up with in the past forced users to choose one of these DPI settings to use. Therefore, it can be interpreted that the average person felt that 400 DPI was too slow, and 1600 DPI was too fast, resulting in 800 DPI being the most common setting.

In game sensitivity is very different. Here it can be observed that pro players will most commonly choose a sensitivity value hovering around 5. As this value increases, the number of players using this sensitivity value drop dramatically, to the point that there are only a handful of professional players that use a sensitivity value of 10 or greater.

As described previously, eDPI is just a multiplication of DPI and Sensitivity. Thus, taking the 2 most common values for both with 800 DPI and 5 sensitivity and a combined value of 4000 is generated. eDPI values between 3000 and 6000 are most common, such that all other combinations have less than half the number of pros using those settings. Finally, the top 5 most used mice by pro players are all Logitech gaming mice. There are several reasons behind why this could be:

1. The team (and therefore the players) are sponsored to use this gear
2. Pros truly think these devices are the best mice available
3. For the G Pro Wireless, it possesses the most neutral design that anyone can use

With multiple metrics, Role and eDPI do not seem to have much influence on each other. This is observed in the following jupyter notebook visualizations below:

Chart, scatter chart

Description automatically generated

Overall, tank and support players have more variance in their eDPI, while damage players are more tightly packed and controlled in their mouse settings. This is illustrated by how there are many players in both the tank and support role that surpass 8000 eDPI but there is not one damage player that has an eDPI of 8000 or more.

Chart, box and whisker chart

Description automatically generated

Tanks appear to have the highest average eDPI with some of the greatest variance as well. Damage and support players have similar average eDPIs but support players' eDPIs are much more varied than that of the damage players.

Chart, bar chart

Description automatically generated

There appears to be a lot of variance in players' eDPIs depending on the hero they are playing. Heroes that require precision aiming typically have the least variance as it is generally known that it is much more difficult to aim when using extremely high eDPIs. Conversely, heroes that have area-of-effect, splash damage, or auto-tracking weapons have a large amount of variance since their hero's ability to perform their designated role is not tied down to their eDPI. This is also true for heroes with defensive abilities/barriers that cover a wide area and benefit from higher eDPIs.

**Transforming skewed data:**Skewed data is created by taking the log of the designated column. Then by taking the skew of the overall dataframe, the outputted value will be between -1 and 1. This generates noticeable changes, particularly in the histogram for Sensitivity and eDPI. These 2 histograms were generated in jupyter notebook and can be seen below:

Chart, histogram

Description automatically generated

Figure : Skewed Data Histogram for Sensitivity

Chart, histogram

Description automatically generated

Figure 2: Skewed Data Histogram for eDPI

BDM500 Project MS5

**Building the predictive model:**Much of this step is going to be in the jupyter notebook file. Having said that, the model built was a logistic regression model. This is because this type of model is best suited for categorical predictions. Since the main objective of this project is to determine the relationship between cmPer360 and Role, this model excels in making a prediction as to what Role a player is depending on all other numerical attributes, including cmPer360.

One important step to mention in this building process would be the .groupby(‘Role’).mean() line. This sorts the information into the averages of each variable/column grouped by their role, it can be determined that Tank players have the highest overall sensitivities as represented by how low the average cmPer360 value is for that role. Damage players are next with a 5cm increase in average cmPer360, with Support players at the top with the slowest sensitivities at a 10cm increase over Tank players in average cmPer360 values.

**Generating the final visualizations:**When looking below at the final logistic regression model plot generated in jupyter notebook that compares cmPer360 and Role, visually it does not appear there exists any correlation between these 2 variables. This claim will be further supported in the next section during metric evaluation.

A picture containing calendar

Description automatically generated

This next visualization was generated in Power BI. Here, a deeper analysis into the Role category is performed with the addition of the Hero category. Based on the average cmPer360, it can be observed that there does not appear to be any discernable system for understanding why some heroes fluctuate so heavily in mouse sensitivity settings, despite many heroes with these differences being a part of the same Role. This is most prevalent in the Damage and Support Roles.

In the case of the Damage Role, Echo is a flying hero, meaning she requires most mouse movements be performed along all planes of movement, most notably vertical movement. As a result, it is expected that she require a low cmPer360 value and she does, having the lowest distance in this dataset to perform a 360 turn. In spite of her character design, Echo is not the only flying hero in the game. Pharah is extremely similar to her as she possesses a jetpack that enables her to take flight as well. Thus, with this overlap in Hero design and team Role, this begs the questions as to why there is such a stark contrast in cmPer360 for these two similar heroes. The answer of course, is that it is not the individual Hero that mandates the mouse settings players use while playing them, but rather what mouse settings the player is comfortable with using while piloting that specific character. Similar to the previous visualization, this claim will be support in the next few pages as the performance metrics are calculated to be used in a concluding statement.  
  
Chart, scatter chart

Description automatically generated

**Evaluate the model using multiple metrics/methods:**

1. First start with accuracy using .score(x\_test, y\_test).



1. Next, create a confusion matrix to better understand the accuracy score metric

A picture containing text

Description automatically generated

* 1. Therefore, the 3 numbers along the main diagonal from top left to bottom right include 2, 8, and 3 for a total of 13. This means there were 13 correct predictions that matched the actual output. The remaining numbers total to 17, leading to a result of 13/(13+17)=0.433.

1. Evaluate 2 more metrics of precision/recall using classification report

A screenshot of a computer

Description automatically generated with medium confidence

* 1. Considering how the best possible value for precision and recall is 1 and the worst is 0, both metrics scored relative to the accuracy metric of this model. It can be observed that for precision, the model had an easier time predicting players belonging to the Support class.
  2. The same phenomenon can be seen for recall as well, where the recall value was highest for successfully identifying players in the Damage role. When scored across all 3 roles, overall, the weighted average swung back around towards the mid 40s range in percentages, with precision at 0.45 and recall at 0.43. Given how the accuracy is 0.43, these metrics are all in line with each other.

1. Finally, use pseudo r-squared using statsmodel summary

A screenshot of a computer

Description automatically generated with medium confidence

* 1. The value of pseudo r-squared is 0.674 and this means that it is on the upper end of having a better model fit than usual. The variable with the greatest coefficient is the cmPer360 variable at 0.02 which indicates this variable is for the most part statistically insignificant. In other words, cmPer360 nor any other variable has much of an impact on what a player's role is.

**Conclusion:**  
Therefore, given what is known about the logistic regression model, it can be concluded that the role of a professional Overwatch player cannot be accurately predicted with this model. Despite using every available numerical variable in this logistic regression model, it is only capable of scoring at about 43% accuracy. This is a surprisingly poor result, but it makes sense considering the quantity of players in the dataset only number in about 150 results. Thus, factor in how across these 150 players, 80% were used for training and 20% were used for testing, and there ends up not being a lot of data for the model to learn from nor test on. Typically, datasets containing thousands of entries are used for machine learning algorithms. As a result, this model does not perform well right now and may never get to the point of being accurate enough to reasonably use. However, improving this model’s baseline accuracy of 43% should be very straightforward as the limitation in the number of results in the dataset is likely playing a large role in hindering this machine learning model.

The overarching reason for why studying this dataset was warranted in the first place hinged on the possibility of there being some configuration of best mouse and mouse settings to use that even professional Overwatch players could unanimously agree on. After performing this analysis, there clearly does not exist any perfect configuration of settings to use when it comes to a player’s mouse or mouse settings. Instead, professional players will typically follow simple guidelines that allow for them to maximize their mouse agility and ease of manipulation, without compromising any degree of control. For instance, pros ensure that their mouse DPI is between 400 and 1600 DPI and only increase their eDPI through the in-game Sensitivity value that matches what they are comfortable with, regardless of what hero or role they play. This applies to the gaming mice professional players use as well since they come in different shapes and sizes, varying in comfortability from player to player.

Ultimately, the key takeaway from this analysis would be to use the mouse and mouse settings that are most comfortable for the individual in question. So long as the mouse is competitively viable (e.g produced by a reputable gaming apparel brand), it should not matter what mouse the player is using, so long as they put in the proper amount of time to be accustomed to using that device. This extends to mouse sensitivity as well, such that copying a professional player’s DPI and in-game sensitivity will not act as a quick fix for a normal player to suddenly become a great player.