

Impact of Gaming Habits and Demographics on Mental Health
CIS 4930: Data Mining Final Project Report
Albert Martinez, Ranzley Bumanglag, Jonathan Guzman, Souhail Marnaoui

I. Introduction

Video games have historically been a popular hobby for many, especially as their main form of stress-relief. It began with William Higinbotham's game "Tennis for Two" in 1958, which involved an analog computer and an oscilloscope screen, allowing two players to simulate a tennis game. Atari's Pong game promptly followed suit in the 1970s, sparking a new flame in the entertainment industry. Soon, the gaming industry became its own entity, evolving over the years and eventually reaching multi-dimensional software. This led to an increase in game complexity, enabling more freedom for game developers, establishing new genres and game styles. While this evolution has streamlined access to gaming as a hobby, there are cases where prolonged exposure caused negative effects on the mind and body. The intent of this project is to create a model to predict a subject's overall mental health outcomes when given information regarding their gaming habits and demographic data such as education level, employment status, age, etc.

II. Literature Survey

Not until recently has research been done regarding the correlation between gaming and mental health. Before then, with video games being one of the first demonstrations of power for the digital age and seen as mostly for children and teens, the media was mostly left alone and did not reach the public eye. Now, with about 3.32 billion "gamers" around the world, it is important to see the impact of gaming habits. The results have been mixed, but there are many positive finds. For example, it was found that video games were "helpful in combating anxiety during the COVID-19 pandemic", helped in "goal-setting behavior", and helped gamers decrease the negative effect of depression by "promoting enjoyment and motivation" [2]. Aside from mental health, research has been done on gaming's effects on relationships and human interaction. In short, there could be some subtle correlations in gaming and relationships: First is that male consumption of violent game content proved to be the best predictor for the existence of attachment issues, the frequency of video game use has a positive correlation with attachment issues, and when either partner views gaming as a relationship issue, attachment issues tend to exist in either partner[3].

Our project will not use the game title column, as those responses contain mainly "League of Legends" and "Other." The most common title found, "League of Legends," is part of the Multiplayer Online Battle Arena (MOBA) genre. There is some research suggesting that prolonged exposure to these types of games could lead to Internet Gaming Disorder (IGD), a temporary disorder recognized by the American Psychiatric Association. [4]

III. Methodology

A. Data Collection

The comprehensive dataset required for this study was procured from the Open Source Framework (OSF), a repository that contains extensive information on individuals' gaming habits and associated mental health indicators. This dataset was meticulously chosen for its breadth of psychometric and demographic variables, including, but not limited to, General Anxiety Disorder (GAD), Satisfaction with Life (SWL), and Social Phobia Inventory (SPIN) scores. Additional data points encompassed age, gender, country of residence, employment status, education level, and specific gaming preferences, which provided a holistic view of the survey participants' profiles.

B. Data Preprocessing

Prior to analysis, the dataset underwent rigorous preprocessing. This multi-step process involved:

- **Data Cleaning:** The initial step was to purge extraneous columns that did not align with the study's objectives, such as timestamps and unrelated survey questions. This was followed by the identification and imputation of missing values to preserve the integrity of the dataset.

- **Variable Transformation:** To ensure all data could be interpreted by the chosen machine learning algorithms, categorical variables were transformed. Nominal variables with no intrinsic order were one-hot encoded, generating binary columns for each category. In contrast, ordinal variables were processed through label encoding to retain their hierarchical nature.

C. Exploratory Data Analysis (EDA)

An extensive EDA was performed to gauge the distribution of mental health scores and to discern potential correlations between variables. This involved:

- **Histograms:** For each mental health score (GAD, SWL, SPIN), histograms were plotted to visualize the frequency distribution and identify skewness or anomalies in the data, and can be viewed from the Jupyter Notebook.

- **Correlation Matrix:** To uncover any linear relationships, a correlation matrix was computed and visualized using a heatmap. This analysis was pivotal in determining intervariable relationships that could influence the subsequent feature selection.

D. Feature Engineering and Dimensionality Reduction

Acknowledging the high dimensionality of the dataset, Principal Component Analysis (PCA) was leveraged for dimensionality reduction. The decision to retain 160 components post-PCA was substantiated by the examination of a scree plot, which demonstrated a plateau in cumulative explained variance, indicating that these components retained the majority of the information present in the original dataset.

E. Model Selection and Training

The study progressed with the development of two distinct machine learning models:

- **Random Forest Classifier:** Chosen for its ensemble approach in handling both bias and variance effectively. A hyperparameter search was conducted to optimize the number of trees and maximum depth for each tree to enhance model performance.

- **Support Vector Machine (SVM):** Implemented with a polynomial kernel to address the non-linear patterns in the data. The degree of the polynomial kernel was a hyperparameter tuned to strike a balance between model complexity and computational feasibility. Each model was meticulously trained to predict the mental health outcomes as reflected by the GAD, SWL, and SPIN scores.

F. Model Evaluation and Validation

Model performance was critically evaluated on the validation set using a suite of metrics:

- **Accuracy:** The primary metric was the overall accuracy, which provided a straightforward assessment of the models' predictive capabilities.

- **Average Distance from True Class:** This novel metric was computed to measure the deviation of the predictions from the actual class labels, thus offering a nuanced understanding of model precision.

- **Class Proximity Analysis:** The analysis extended to calculate the percentage of predictions that fell within two, three, and five classes from the true label, furnishing a granular view of the models' predictive accuracy across the spectrum of the outcome classes.

The evaluation results revealed that the models demonstrated modest accuracy, suggesting the need for further refinement in feature engineering, model selection, or both.

G. Iterative Refinement

Given the initial results, an iterative refinement process was employed. This included revisiting the preprocessing steps, experimenting with different hyperparameters, and considering alternative modeling techniques such as ensemble methods or deep learning to potentially enhance predictive performance.

IV. Implementation

A. Basic cleaning and target selection

Our data was initially gathered as a .CSV (comma-separated value) file to be read in as a DataFrame object with 13,464 rows and 55 columns. Columns would be dropped so that the model was only fed meaningful data pertaining to gaming habits and demographic information. Columns representing mental health information would be designated as the target variables. GAD_T, SWL_T, SPIN_T represent the aggregate score of all the questions related to General Anxiety Disorder, Satisfaction with Life, and Social Phobia Inventory, each of which had their own separate column, and were therefore chosen as our three target variables while also dropping the columns representing the individual questions. These targets were chosen for their faithful representations of mental health information, and for their recognition by the American Psychiatric Association as valid measurements of mental health outcomes. The following columns were dropped for the following reasons:

- 'Highestleague' - (Overwhelming amount of custom responses)
- 'League' - (Overwhelming amount of custom responses)
- 'Zeitstempel' - (Not meaningful to the model's purposes)
- 'GADE' - (Includes mental health information)
- 'Game' - (Overwhelming amount of custom responses)
- 'League' - (Overwhelming amount of custom responses)
- 'Narcissism' - (Self reported, non-standardized mental health information)
- 'Reference' - (Not meaningful to the model's purposes)
- 'Accept' - (Not meaningful to the model's purposes)
- 'Residence' - (Overwhelming amount of custom responses)
- 'Birthplace' - (Overwhelming amount of custom responses)

Certain columns needed cleaning, but were kept due to the valuable nature of the information and their manageability. These were deemed manageable due to the amount of samples containing custom responses in these columns being negligible, meaning they could be imputed without losing a significant amount of data. The columns listed below fell in this category:

- 'earnings'
- 'whyplay'
- 'Playstyle'

B. Encoding

Nominal categorical information for information like gaming platform, gender, or playstyle was encoded using the Pandas library's `get_dummies()` function, while Ordinal encoding for columns like "Degree" was done using `sklearn.preprocessing.LabelEncoder()`. Attributes containing custom entries were marked as missing attributes or, NaN.

C. Dimensionality Reduction

Encoding caused a significant increase in dimensions, jumping from 43 attributes to 251. We decided to employ a dimensionality reduction algorithm, and decided on a Principal Component Analysis (PCA) algorithm. We used the module 'PCA' provided in the `sklearn.decomposition` library.

During PCA dimensionality reduction, we would have to remove all samples with missing data due to the `sklearn.decomposition.PCA` algorithm having a zero-tolerance for NaN entries. NaN stands for "Not a number" and is used to represent missing data, or data that cannot be meaningfully represented for the model to use.

Removing the samples would only be a viable option if it did not remove a significant portion of our dataset. After dropping all samples with missing data, approximately 762 samples were removed. Given that this was a small portion out of the 13,464 total samples, it can be considered a satisfactory trade off for being able to implement dimensionality reduction.

E. Classification

We used the Random Forest and SVM classifiers from the sklearn library to train our models. The polynomial kernel was used to train our Support Vector Machines, and the Random Forest models were instructed to create 100 estimators (trees) per model. Additionally, following the poor initial performance of our SVM model, we reimplemented it using a OneVsRestClassifier provided in the sklearn library in order to combat SVM's weakness against large-scale multi-class classification.

F. Tools + Additional Notes about

Jupyter Notebook was used to write and run our python code, and the following python libraries were used: Pandas, seaborn, matplotlib, numpy, sklearn. We additionally attempted to stratify the data during the test split to ensure an even class distribution among the training and validation data, but this proved difficult given the nature of our data. There are extremely few classes in our data that are represented by only one sample, and therefore cause an error when that sample cannot be split among the two sets. This prevented the attempt for intelligent splitting of our data, and we instead opted to keep our random training and validation splitting method.

V. Results

```
In [25]: """ Evaluate the random forest model performances """
#ax = plt.gca()

for target, model in RF_models.items():
    y_pred = model.predict(validation_features)
    accuracy = accuracy_score(validation_targets[target], y_pred)
    # conf_matrix = confusion_matrix(validation_targets[target], y_pred)
    print(f"Accuracy for {target}: {accuracy}")

Accuracy for GAD_T: 0.35143644234553323
Accuracy for SWL_T: 0.19874065328610782
Accuracy for SPIN_T: 0.10350255804801259
```

```
In [26]: """ Evaluate the SVM model performances """
ax = plt.gca()

for target, model in SVM_models.items():
    y_pred = model.predict(validation_features)
    accuracy = accuracy_score(validation_targets[target], y_pred)
    # conf_matrix = confusion_matrix(validation_targets[target], y_pred)
    print(f"Accuracy for {target}: {accuracy}")

Accuracy for GAD_T: 0.12632821723730814
Accuracy for SWL_T: 0.04132231404958678
Accuracy for SPIN_T: 0.03699330972058245
```

Above: The first performance evaluations of the SVM models and the Random Forest models. Our highest performing model was a Random Forest that had an accuracy of 35% on the GAD_T target, and our lowest performing model was an SVM that had an accuracy of 3% on the SPIN_T target.

As you can see, they performed very poorly. Should even a random guesser at least be capable of predicting with a 50% rate of accuracy? Upon recalling the nature of our data, we can tell that this would not quite be the case. Our dataset is non-binary, which means that the targets boast many possible classes, and each sample can only belong to a single one of them.

GAD_T: 22 classes, represented by integers 0-21

SWL_T: 31 classes, represented by integers 0-30

SPIN_T: 67 classes, represented by integers 0-66

For random guessers trying to guess which singular number classifies a sample from GAD_T, the odds of success are 1/22. For SWL_T, the odds are 1/31. For SPIN_T, they are 1/67. With this in mind, we measured the rate of accuracy of our models vs our theoretical random guesser, rounded to the nearest ten-thousandth.

	GAD_T	SWL_T	SPIN_T
RF	0.3514	0.1987	0.1035
SVM	0.1263	0.0413	0.0370
Random Guesser	0.04545 (1/22)	0.032258 (1/31)	0.014925 (1/67)

Both our models perform better than a random guesser, in particular our random forest model, but the results are still less than satisfactory since there is little use for a predictive model that is, on average, correct only 21.79% of the time.

Our SVM model performs particularly poorly. We first suspected this may have to do with the fact that it is encountering multi-class classification on a large scale. If it could classify each of the 0-66 possible classes individually as their own binary label, and choose the one with the highest degree of confidence, it may be able to make more accurate predictions. Towards that end, we attempted running the SVM model with a One vs. Rest classifier, then re-evaluating its new score.

```
In [29]: """ Evaluate the SVM model performances """
ax = plt.gca()

for target, model in SVM_models.items():
    y_pred = model.predict(validation_features)
    accuracy = accuracy_score(validation_targets[target], y_pred)
    # conf_matrix = confusion_matrix(validation_targets[target], y_pred)

    print(f"Accuracy for {target}: {accuracy}")

Accuracy for GAD_T: 0.24714679260133807
Accuracy for SWL_T: 0.1349862258953168
Accuracy for SPIN_T: 0.0747737111373475
```

Our SVM model is now seeing at least double the increase in performance on the GAD_T and SPIN_T targets, and triple the performance on the SWL_T target. Here are the updated statistics, once again rounded to the nearest ten-thousandth:

	GAD_T	SWL_T	SPIN_T
RF	0.3514	0.1987	0.1035
SVM	0.2471 0.1263	0.1350 0.0413	0.0748 0.0370
Random Guesser	0.0455	0.0323	0.0149

Our SVM model has improved, but our overall scores of 24%, 13%, and 7% are still unsatisfactory, and our random forest models are not performing significantly better. However, we suspected that we may have reason to believe that the accuracy measurement provided by the sklearn library isn't faithfully representing the ability of our models.

At this point, it is important to realize that the `accuracy_score()` algorithm is only counting perfect predictions as a success, and anything else as a failure. This constitutes a problem in our measurements due to a loss of information.

For instance, consider all of the classes in target `SPIN_T`, which range from 0-66, and all of which have an ordinal relationship. If the true class of a sample was 56, and the model predicted 55, that shows an ability to predict remarkably close to the true class, but it is still counted as a complete failure in our accuracy measurements.. The reason this is considered a form of misrepresentation is because a close, though not exact, prediction to the true class has semantic meaning in this dataset and should therefore be represented in some alternative accuracy measure in addition to the strict accuracy measurement provided by the `sklearn` library.

At this point we must devise our own measurement of accuracy that will provide meaningful insight into the capability of our models. First, we can count all guesses that fall within a certain distance of the true class to be correct, and all that fall out to be incorrect. With this measure, we can experiment setting different values for the distance, and see how our model fares with different thresholds. Another measure involves computing the average distance from the true class among all predictions. In order for this to have meaning, we must consider that the distance between a random guesser's guesses and the true class on all samples would reach an average distance equal to half of the class range. For instance, the average distance of a random guesser's guess from the true class on target `SPIN_T`, whose classes are 0-66 would be approximately 33. Before we implement this, let us re-fit and re-split the model to observe variance in scores, while providing additional metrics such as precision, recall, and F1 score.

Below: Performance re-evaluations of the SVM (Left) and Random Forest models (Right) after re-splitting and re-fitting, with basic measures of precision, recall, and F1 score added.

Accuracy for GAD_T: 0.2715466351829988 Precision for GAD_T: 0.1649556850494367 Recall for GAD_T: 0.17946973445801803 F1 for GAD_T: 0.16473559556165426 Accuracy for SWL_T: 0.12908303817394726 Precision for SWL_T: 0.13703558040707478 Recall for SWL_T: 0.14722041963934068 F1 for SWL_T: 0.13279307293294937 Accuracy for SPIN_T: 0.06650924832743015 Precision for SPIN_T: 0.05111556246603666 Recall for SPIN_T: 0.04687051216167883 F1 for SPIN_T: 0.04649700720738474	Accuracy for GAD_T: 0.3900039354584809 Precision for GAD_T: 0.34175962725459663 Recall for GAD_T: 0.29184110130360563 F1 for GAD_T: 0.2984937772265575 Accuracy for SWL_T: 0.1940181031090122 Precision for SWL_T: 0.21244783986470223 Recall for SWL_T: 0.18565440454347534 F1 for SWL_T: 0.19232799632367312 Accuracy for SPIN_T: 0.1058638331365604 Precision for SPIN_T: 0.08954901185678626 Recall for SPIN_T: 0.07911152284255914 F1 for SPIN_T: 0.08180871973890068
---	---

When re-evaluating using the average distance measure, as well the distance threshold measure, we get considerably more promising results. Please note that the majority of the details reported below are omitted for the sake of preserving readability of this report, and can be read in their entirety when you run the 12th and 13th cells on the Jupyter Notebook.

RF Model average dist from all targets: [2.5799554] - Random guesser would be: 20.7

SVM Model average dist from all targets: [5.51843106] - Random guesser would be: 20.7

% of RF predictions landing within 2 classes of the GAD_T true class: 86.30%

% of SVM predictions landing within 2 classes of the GAD_T true class: 67.02%

VI. Conclusions

Our original intent was to find the most influential factor in an individual's mental health. However, as research progressed, the problem evolved into predicting the status of a test sample's mental health scores across the three different psychological evaluations (GAD, SWL, and SPIN). Finding the most influential factor in a specified context (gaming in this case) would not be as beneficial as creating an algorithm that can be applied to various data sets. Thus, we decided to employ the random forest model (RF) to predict mental health scores. With regards to the model, from our testing, we discovered that the random forest algorithm proved far more effective relative to the support vector machine model (SVM), and both were more effective than a random guesser. The only limiter to this algorithm would be the fact that we did not consider rank. Rank affects hours played and the motivation for playing, both of which could in turn affect mental health in the long turn.

VII. Future Research

As mentioned in the previous section, rank limits the potential for the algorithm's accuracy. In order to analyze rank across various games, the rank tiers must be standardized into a single scale (likely percentile). In competitive multiplayer games, players can obtain a rank as a measure of skill relative to other players. The next step in analyzing the effect of gaming habits and demographics would be how rank correlates with stress levels. To climb the ranks of a competitive game, more dedication is required to maintain the respective rank's skill level. This impacts hours played, and even the motivation for playing. Once a player reaches a certain skill ceiling, the motivation starts to shift from a casual one to a more serious, maybe even semi-professional attitude. Consequently, what began as a stress-reliever could potentially change into a stress-inducer. A possible method of analysis would be to use the stress survey data to establish a model that predicts rank, assuming the games in question have a ranking system.

VIII. References

- [1] M. Sauter and D. Draschkow, "Gaming Habits and Psychological Well-being: An international dataset about the Anxiety, Life Satisfaction and Social Phobia of over 13000 gamers," *OSF*, Oct. 2017.
- [2] M. Kowal , E. Conroy , N. Ramsbottom , T. Smithies, A. Toth , M. Campbell, "Gaming Your Mental Health: A Narrative Review on Mitigating Symptoms of Depression and Anxiety Using Commercial Video Games" , *JMIR Publications* , June. 2021.
- [3] James McClellan Smith, "The Relationship Between Video Game Use and Couple Attachment Behaviors in Committed Romantic Relationships Attachment Behaviors in Committed Romantic Relationships", Brigham Young University , June 2013.
- [4] S. Park, J. Ha, W.Ahn, and L. Kim, "Measurement of craving among gamers with internet gaming disorder Using repeated presentations of game videos: a resting-state electroencephalography study," *BMC Public Health*, Vol. 23, no. 1, May 2023