COMS30050: Applied Data Science

Mental Health and Screen Time

Jason Park (jp17528), Sungjin Kang (ou18496), Tom Stark (jb18789),
Junoh Park (ov18075), Ben Lee (ex18124)

I. Abstract

With a rise in the consumption of digital media in the past decade, there has been a big focus on the effect this could have on mental health, especially teenagers and young adults. Specifically this report outlines our approach to answering the question

"Does the screen use at age 16 affect the level of depression and anxiety at age 18?"

Given a synthetic dataset based on data collected from the **ALSPAC** (Avon Longitudinal Study of Parents and Children) cohort study data set, we extracted variables that directly related to screen time and target variables that directly linked to depression and anxiety. A lot of the variable values within the dataset were categorical/interval so we came up with a numerical classification method to standardise our analysis.

We used binary classification to classify our 6 target variables: `dep_score`, `dep_thoughts`, `has_dep_diag`, `prim_diag`, `secd_diag`, `panic_score` to either indicate depression or no depression.

By displaying our data in a correlation matrix using Spearman correlation, we have discovered that the higher someones screen time during the week, the higher their screen time on the weekend. It is unlikely to have the correlation between our data and our target variables. We then tried to find a correlation with further techniques like PCA.

We selected four different machine learning algorithms to try and model a relationship between screen time and depression and anxiety using accuracy and F1-score to evaluate the performance.

While different algorithms had different success in modelling depression and anxiety from screen time, we cannot say for certain that we can predict mental health from screen time due to the accuracy of our algorithms not being high enough. We did however achieve an accuracy higher than if we were to randomly guess depression and anxiety.

This was most likely due to us not having enough data to work with and our dataset being based on outdated 'screen time' data.

II. Introduction

A. The rise of the digital age

The rise of digital media has taken over our lives in the last decade with projections to continue doing so with a 9% rise in social media users year-on-year[14]. This comes with many benefits on society, connecting us in more ways than ever and giving people a way to communicate like never before. This was particularly evident during the national lockdowns of the COVID-19 pandemic. Although we were physically apart from each other, we were more digitally connected than ever.

With the smartphone coming to market in 2007, it quickly became the main source of interaction with every online service that we use, and by 2015 92% of teenagers and young adults owned a smartphone[24]. Coincidentally this correlated with the rise in depressive symptoms in teenagers and young adults over the same time, even when matched year-by-year. Over the same period there was also a sharp rise in reports of students seeking help at university counselling services, with the main reasons being for anxiety and depression.

B. Impact on mental health

In particular the younger generation, who have grown up with a dominating social media presence, have been among the most affected. Social interaction is essential for someones mental health and well-being but the use of social media means people spend much less time connecting with people in person. Instead they spend more time socialising online through social media. Interacting with people in this digital way not only leaves users feeling less emotionally satisfied than if they had formed those connections in person; but it also can leave you feeling very negative about yourself, feeling envious or jealous of other peoples unrealistic portrayals of their lives which can then lead to depression and anxiety.

This project aims to outline our findings for the following question

Does the screen use at age 16 affect the level of depression and anxiety at age 18?

C. The dataset

The data we used is a synthetic dataset that has been specifically created for this question. It was synthesised from a subset of the **ALSPC** cohort dataset[15] which has been following a set of children since birth, monitoring certain aspects of their life.

D. Possible limitations

It has become increasingly obvious that social media and screen time can have a detrimental effect on our mental health. The timeline over which our data is based on, it may not present this as strongly as we hoped.

The **ALSPC** followed children born in 1992, making it 2008 and 2010 when they were 16 and 18. As previously mentioned, the smartphone was only announced in 2007 and it still took a while for it to saturate the younger demographic. Therefore, the data collected (or synthesised) for screen usage in the children from this study might be vastly different from what you may expect from children who turned 16 and 18 in the past couple of years.

III. Data Preparation

Before defining any models, we needed to explore our dataset and remove any unrelated, biased or polluting data that may significantly affect the effectiveness of our models.

A. Ethical concerns

The dataset we are using has purely synthetic data values which means in theory there is no need for us to be particularly careful with how we use the data and which variables we select to do our data analysis. However, the problem is that this data could easily be real, especially since it was based off real data collected from real children.

The question given to us to answer has also already been analysed with real data and therefore it is only right for us to have a look at the privacy and ethical concerns of our dataset. The dataset provides 82 variables of which none are direct *identifiers*.

There are a handful of columns that are *quasi-identifiers* namely `sex` which identifies the individuals gender. Every variable also has a label for the age at which the data was taken. While this does not directly identify an individuals age there is a concern

that an individuals value for this variable (at a given age), could be linked using an external non-anonymous dataset.

The main source of confidential information found in our dataset comes from depression and anxiety diagnoses. To deal with this we have suppressed any *quasi-identifiers* by not including them in our data or target variables. Although one of our target variables `has_dep_diag` does include confidential information, the exclusion of all *quasi-identifiers* makes our analysis anonymous.

B. Categorising columns

An initial observation of our dataset helped identify any unrelated columns. Since we only needed data relating to screen time at age 16 and the level of depression and anxiety information at age 18, we could eliminate any columns that did not match that criteria. For example, some columns of data collected were collected at age 5.

We labelled a column *DS* if it is directly related to the screen time (for example `comp_week`), *IS* for columns indirectly relating to the screen time, *DM* for columns directly relating to the mental health (depression and anxiety), and *IM* for columns indirectly relating to the mental health. We focus on the following columns: *DS* and *DM* since these were the two we wanted to find a relationship between.

C. Decide Targets

Next, we needed to decide which columns to select relating to screen time as our inputs, $\mathbf{X}$, and which columns to select relating to depression and anxiety as our targets, $\mathbf{y}$. Our target variables are shown in the table below.

Column Name	Definition
`dep_score`	Child's depression score on CIS-R.
`dep_thoughts`	Child's number of depressive thoughts on CIS-R.
`has_dep_diag`	Child has ICD-10 diagnosis of depression.
`prim_diag`	Child's primary diagnosis in accordance with ICD-10 criteria.
`secd_diag`	Child's secondary diagnosis in accordance with ICD-10.
`panic_score`	Child's panic score on CIS-R.

Table. I: Target columns with its definition

We chose these variables as they directly quantified/categorised the level of depression and anxiety.

D. Classify values

Some of the variables that we chose were not numerical. For example, the column `comp_week` represents the average time a child spent per day using a computer on a typical weekday and was classified by 'Not at all', 'Less than 1 hour', '1-2 hours' and '3 or more hours'. Since our machine learning models only accepted numerical data we had to convert them to an integer scale, setting 'Not at all' to 0, 'Less than 1 hour' to 1, '1-2 hours' to 2 and '3 or more hours' to 3. We followed a similar approach for the rest of the selected variables.

We first tried to combine all of our targets, however this was not possible as the quantification of depression was different for each variable. For example, having a `dep_score` of 3 and `prim_diag` are different things. The former is just a score for depression whereas the latter represented ICD-10 criteria. In addition, `has_dep_diag` only has values 1 and 0, representing having a ICD-10 diagnosis or not respectively. Instead, we decided to find a relation between the input variables $\mathbf{X}$ and each individual target.

E. Exploratory Data Analysis

We started our analysis with Linear Regression which is popular method that models the relationship between a target variable and one or more explanatory variables. These relationships are modeled using linear prediction functions who's model parameters are estimated from the input data. It aims to give the best fitting line that minimises the residual defined by the *Mean Squared Error*.

We chose one input variable X, `comp_week`, and ran Linear Regression with one target, `dep_score`, and plotted a graph. The objective of a linear regression model is to find a continuous relationship between the input variables and a target. The main reason we tried this approach was because it could give us a general idea of how to approach manipulating our dataset:

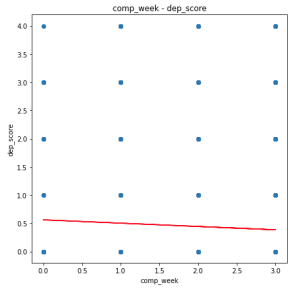

Fig. 1: Graph of Linear Regression with `comp_week` and `dep_score`. The blue dots are overlapping each other showing only 20 dots. If there was a positive relationship you might see a graph that is similar to $y = x$.

The above graph showed that this attempt was not the correct approach for our dataset but it gave us some useful information. Firstly, we noticed that we should perform 'classification' rather than 'regression' because the distribution of $\mathbf{y}$ targets are categorical and not continuous.

Secondly, as you can see from the graph, the line that is drawn is very low down. The linear regression model has estimated that any value of `comp_week` would only have a `dep_score` between 0 and 1. From this graph we realised that the distribution of our data might not be uniform; therefore, we decided to check the distributions of all our $\mathbf{y}$ targets.

By looking at Fig. [11], you can clearly see that the distribution of all our targets is not uniform.

F. NaN

There were lots of NaN values present throughout our data. There are two common approaches for dealing with NaN values: We could either predict them based on the rest of the dataset with machine learning, or we could just delete them. Of our chosen targets `dep_thoughts` only had 1254 values (10%) out of 13734 that were not NaN. The other targets only had 3043 non-NaN values (22%). Since the size of our non-NaN dataset was not big enough to predict the NaN values, we opted to simply remove any NaN rows.

G. Final data categorisation

We categorised our target values into either 0 or 1: 0 for no depression or anxiety and 1 for having depression or anxiety, leaving us with a binary classification of our target variables. There were several reasons for doing this. Firstly, we found that it is very common for a person with either anxiety or depression also has the other[36, 21, 9, 27]. Therefore, we felt it was reasonable to unify variables that indicated depression or anxiety. Secondly, our target $\mathbf{y}$ values did not have the same meaning. For example, `dep_score` and `dep_thought` were classified between 0 and 4 showing the strength of the depression, but

`prim_diag` and `secd_diag` were classified from 0 to 12 which represents the name of mental health disease. Mapping their values to 0 and 1 helped to standardise our findings.

In addition, we created a classification column for the total level of depression and anxiety (See Fig. 2). We summed the values of our 6 targets. This gave us an overall depression and anxiety classification between 0 and 6. From this, we could predict how likely a person is to have depression or anxiety while also predicting how serious their depression or anxiety may be.

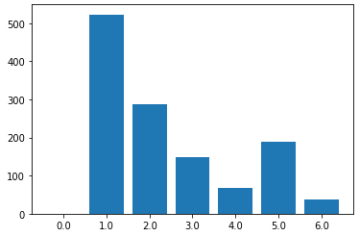

Fig. 2: Distribution of Combined y Targets (classification). The distribution is imbalanced.

H. Sampling the data - Under-sampling, Over-sampling

For the final step in our data preparation, we created a sampled (under-sampling and over-sampling) dataset to try and eliminate the imbalance we previously discovered in our data.

For example, in `dep_thouhts` we had 3043 entries of which 2339 were classified as 0 and 704 were classified as 1. This meant that before making any predictions with a machine learning model, we would already have an accuracy of 76.9% without doing any analysis if we classified a new data point as 0.

To get around this problem, we randomly under-sampled[3] an equal amount of rows of both 0 and 1 classifications leaving us in the case of `dep_thoughts` with 704 classifications of both 0 and 1. This unfortunately did further reduce our data size but it left us with data that wasn't imbalanced, which we thought was more important.

We also randomly over-sampled dataset due to the lack of original data we had for under-sampling. For our combined y targets we also performed under-sampled and over-sampled.

IV. DATA EXPLORATION

After we had prepared our data to properly measure both screen time (our $\mathbf{X}$s) and mental health ($\mathbf{y}$s), we visualised and explored the relationship between them. We used 3 different methods for data exploration and visualisation: Spearman's correlation coefficient and p-value, PCA with kernels, and t-SNE.

A. Correlation Matrix and P-value

We hoped that using a correlation matrix would provide some insight into the relationships between our variables[12]. A positive or negative correlation would result in a value of near 1 or -1 respectively and a weak relationship would be represented by a value close to 0.

We used Spearman's correlation over Pearson's correlation. Firstly, because our $\mathbf{X}$s and $\mathbf{y}$s are not normally distributed, we found this out by performing a normality check using the Shapiro-Wilk test [29] for each variable. Secondly, we are measuring the relationship between two *ordinal* variables.

Hauke and Kossowski [17], make it clear that there is an obvious difference between Pearson's and Spearman's correlation; thus, choosing an appropriate method was crucial in giving us a valid interpretation. The equation for calculating Spearman's correlation between two data can be expressed like this:

$$cor(P,Q) = \frac{\sum P_i Q_i - \frac{(\sum P_i)(\sum Q_i)}{n}}{\sqrt{\sum {P_i}^2 - \frac{(\sum P_i)^2}{n}}\sqrt{\sum {Q_i}^2 - \frac{(\sum Q_i)^2}{n}}} \quad (1)$$

Calculating the p-values between $\mathbf{X}$s and $\mathbf{y}$s also provides insight on the correlation in our data as it measures the significance of $cor(X,Y)$. The more significant a correlation between two variables, the smaller the p-value. If the p-value is smaller than or equal to 0.05, it is said to be statistically significant and therefore, the corresponding correlation coefficient is meaningful. By observing the correlation matrices in Fig. 3, we can observe

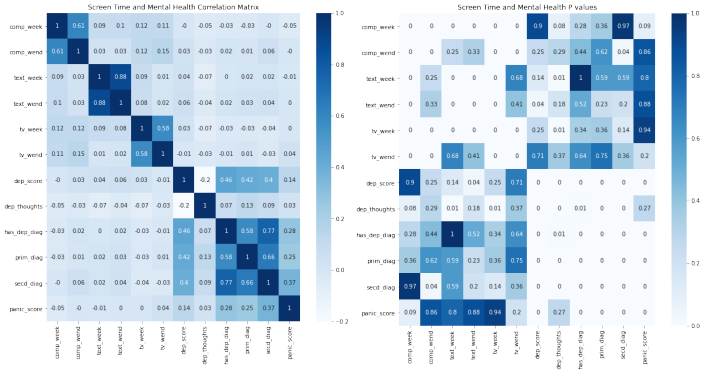

Fig. 3: **Left**: Correlation matrix between screen time and mental health variables with total data-set. The darker the cell, the higher the correlation between two variables, the lighter the cell the less correlation there is. **Right**: Correlation p-value matrix between our screen time and mental health variables. Here, the lighter the cell, the more the significant the corresponding correlation value is. See Fig. [7] for sampled version.

a few notable things.

- **Left**: The longer someone spends on the computer, texting, or watching tv during the week, the longer they will spend doing the same on the weekend.
- **Left**: There is nearly no relationship between our selected $\mathbf{X}$s and $\mathbf{y}$s since the correlation value of each cell (apart from the diagonal) is very close to 0.
- **Left**: There are some notable relationships between our selected $\mathbf{y}$ values. The bottom right hand corner of the correlation matrix is a lot darker relative to the other regions.
- **Right**: Top right and bottom left corner of the p-value correlation matrix is comparably darker than the other regions. This indicates that correlation coefficients in that region are generally statistically insignificant. In other words, we cannot trust these correlations as much due to lack of significance. This is important to note because it implies that the correlation coefficients cannot sufficiently explain the relationship between our $\mathbf{X}$s and $\mathbf{y}$s.
- **Right**: Bottom right corner of the heat-map is relatively lighter than the other regions. This indicates that the correlation coefficients in this region are statistically significant meaning we can trust the correlation values (See the relationship between $\mathbf{y}$s). This also implies that our data wrangling process was reasonable.
- **Right**: There is comparably little relationship between `dep_thoughts` and our other y values.

B. PCA with Kernels

We visualised the relationship between our $\mathbf{X}$s and each y in 2D by using Principal Component Analysis (PCA) and Kernel PCA

(KPCA) [28]. These are both widely used techniques for data exploration and visualisation as it reduces the number of features to consider in relation to the target y.

For KPCA to give us the result we want, it should be able to draw a boundary to divide depression *D* and no depression *ND*. In other words, we should not see much overlap between *D* and *ND* in the 2D scattered plot.

The PCA algorithm transforms the original (high-dimensional) data into low-dimensional data by finding the principal axis that preserves the variance of high-dimensional data as much as possible. The principal axis can be found through feature selection by calculating the eigenvalue of the original data. The higher the standard deviation of the specific feature, the more likely it is to be selected as a principal component.

One of the major drawbacks of PCA however is it cannot be used for non-linear data. This is where KPCA can be really useful because it can be used on non-linear data. Therefore, we opted to use KPCA as well as PCA since it possibly provides better intuition than normal PCA. We experimented with all kernels provided by `scikit-learn` to see which one gave us the best result.

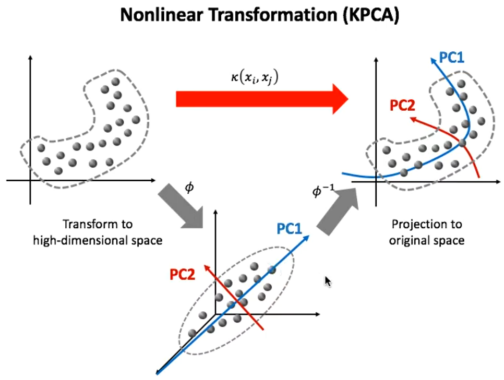

Fig. 4: Explanation of Kernel PCA [4]. $k(x_i, x_i)$ denotes a kernel function. ϕ denotes transformation from lower dimension to higher dimension. There are only two principal components because we try to visualise our data into 2D.

In KPCA, the principal component can be found by performing PCA after converting (ϕ) the data from the original space into a feature space, in which the principal component is linear. However, it becomes non-linear when it is converted back (ϕ^{-1}) to the original space. The kernel makes this process possible without converting the data into higher dimensions (See Fig. 4). This is called 'kernel trick'. It is possible to find a non-linear axis that can describe the original data by utilising KPCA and kernel trick. The following steps describe how to perform KPCA [25]. The details of each kernel function has been omitted and can be found in the paper by Ezukwoke and Zareian's[13].

1) Pick a kernel function: $k(x_i, x_j)$
2) Calculate the kernel matrix: $K = k(x_i, x_j)$
3) Normalise K by using the following equation: $\widetilde{K} = K - 2 \cdot 1_{1/n} K + 1_{1/n} K 1_{1/n}$. n denotes the number of rows of data. $1_{1/n}$ denotes n by n matrix with entries $\frac{1}{n}$.
4) Solve for eigenvalues: $\widetilde{K}\alpha_i = \lambda_i \alpha_i$. α_i denotes ith eigenvector and λ_i denotes ith eigenvalue.
5) Project the data to each new dimension D: $y_D = \sum_{i=1}^{n} \alpha_{iD} k(x_i, x)$ for $D = 1, 2, \cdots$. Since we are projecting the data into 2D, $D = 2$.

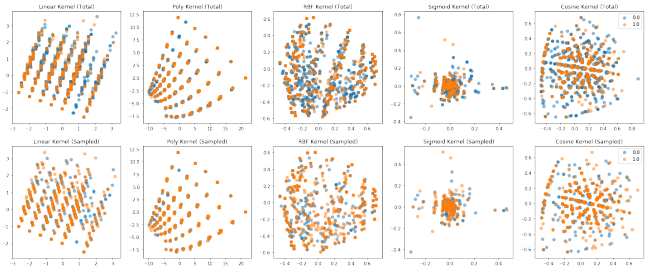

Fig. 5: Top: Total data. Bottom: Sampled data. This figure shows the relationship between the Xs and the y target `dep_score`. The blue dots indicate no depression and orange indicate depression. The 'linear' kernel is the same as normal PCA. See [26] for more results.

Fig. 5 shows that there is no relationship between Xs and `dep_score`. Similar results are observed with all other y targets.

C. t-SNE

Similar to KPCA, the t-SNE algorithm [35] is a non-linear dimensional reduction and data visualisation technique. t-SNE is generally considered a better method than KPCA because it tends to preserve the local and global structure of the data[22]. We hoped that this property of t-SNE would provide better visualisation than KPCA. If t-SNE works well, it should show scattered dots for depression and no depression without them being collated or overlapping.

t-SNE generates an initial solution from the normal distribution $\mathcal{N}(0, 10^{-4}I)$. It iteratively updates the low dimensional solution Q by considering the relations between the samples in higher dimensions P, with gradient decent using a given cost function C. The cost can be measured by KL divergence [23] where large penalties are given when nearby data in P is represented separately in Q; whereas, it hardly penalises when widely separated data in P is represented closely in Q. These explanations can be expressed by the following equations.

$$C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ji}} \tag{2}$$

where p_{ij} and q_{ij} are

$$p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n} \tag{3}$$

$$q_{ij} = \frac{(1 + ||y_i - y_j||^2)^{-1}}{\sum_{k \neq l}(1 + ||y_k - y_l||^2)^{-1}} \tag{4}$$

where

$$p_{j|i} = \frac{e^{-\frac{||x_i - x_j||^2}{2{\sigma_i}^2}}}{\sum_{k \neq i} e^{-\frac{||x_i - x_j||^2}{2{\sigma_i}^2}}} \tag{5}$$

$p_{j|i}$ is a conditional probability of j given i. The closer the distance between i and j, the greater the $p_{j|i}$. A similar rule could be applied to $p_{i|j}$. x_i and y_i denotes i-th sample in P and Q respectively. ${\sigma_i}^2$ is called the 'perplexity' and is one of the hyperparameters for t-SNE. It can be expressed by the following equation.

$$perplexity(P_i) = 2^E = 2^{-\sum_j p_{j|i} \log_2 p_{j|i}} \tag{6}$$

E denotes the entropy of the probability distribution P. For example, if $perplexity(P_i) = 30$ then the variance is calculated for each sample to match the $perplexity$ value.

The solution gets updated by the following rule.

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)(1 + ||y_i - y_j||^2)^{-1} \tag{7}$$

$$y^t = y^{t-1} + \eta \frac{\delta C}{\delta y} + m(t)(y^{t-1} - y^{t-2}) \quad (8)$$

where $y^t = \{y_1, y_2, ..., y_n\}$, η denotes the learning rate, and $m(t)$ denotes the momentum at step t.

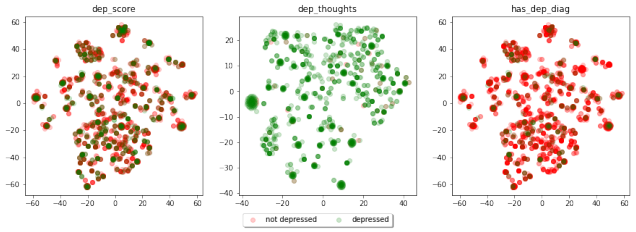

Fig. 6: Results after performing t-SNE on Xs and the first 3 ys on the total version of the data-set. The hyperparameters for t-SNE were left default as set by `scikit-learn`. See [32] for the full plot and more results.

Fig. 6 shows that there is no relationship between Xs and each depression or anxiety related variable y in the total version of the dataset. Similar results were observed for the sampled dataset as well.

There are additional visualisation approaches that we could have explored such as Multi-Dimensional Scaling (MDS) and Isomap. However, we decided to use machine learning algorithms instead of these because they are unlikely to provide better intuition than KPCA and t-SNE.

V. DATA MODELLING [6]

In the previous section, we used various methods to visualise and explore the dataset. For each method, we were unable to find any notable relationship between screen-time and depression or anxiety. Our next step was to try and model our dataset with 4 different machine learning algorithms to see how accurately we were able to predict depression or anxiety from screen time.

We used Support Vector Machine (SVM), Logistic Regression, K-Nearest Neighbors, and Random Forest. We used our total, under-sampled, and over-sampled datasets to see how these models would be affected given balanced (sampled) and imbalanced (total) data.

Since we were trying to predict each of our 6 depression and anxiety targets separately, we modelled each target with each machine learning algorithm. We then combined our targets to give a 'total' *level* of depression between 0 and 6. We used two metrics to evaluate the performance of our model: accuracy and F1 score. For all algorithms, we first performed a hyperparameter search (See [8]) using 3 different seeds. This gave us 3 training and validation curves from which we could generate an average and a standard error for more robust results.

The training and validation curves converge and the validation accuracy increases when the learning is working well. The validation accuracy remains constant or highly fluctuates when the learning is not working well. We also used k-fold cross validation with k=5.

A. SVM

SVM is a method for classification tasks which generates a decision boundary from training data then decides on a boundary for new input data. There are two types of decision boundaries: linearly separable and linearly non-separable. If the data can be separated by a linear line, it is linearly separable. If the data cannot be separated by a linear line or if a kernel is used, then the data is classified as linearly non-separable [19]. SVM has a disadvantage for interpretation of variable weights and individual impact as it directly gives a resulting class without any probability. However, giving a direct result also shows that SVM is a good option when there is no prior knowledge of the data.

1) Algorithm: SVM generates a hyperplane and use it as a decision boundary. From a geometrical point of view of linearly separable, SVM aims to construct the hyperplane to maximise the margin which represents distance between the data and the hyperplane. Given the classifier function:

$$h(x) = g(w^T x + b) \quad (9)$$

the classifier function is combined with the functional margin:

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x + b) \quad (10)$$

By combining equation 9 and 10, we can derive the maximisation of the distance as follows:

$$max_{r,w,b} = \gamma \quad (11)$$

subject to

$$y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \ ||w|| = 1 \quad (12)$$

This means that all distances between the hyperplane and the individual data are more than the length of the maximum distance equivalent to the definition of SVM. Converting the above formula:

$$max_{r,w,b} = \frac{\gamma}{||w||} \quad (13)$$

$$y^{(i)}(w^T x^{(i)} + b) \geq \gamma \quad (14)$$

Here, we apply the scaling constraint $\hat{\gamma}$ because incrementing w and b could increase the margin as if it is getting larger. By setting constraint as $\hat{\gamma} = 1$, $\frac{\hat{\gamma}}{||w||} = \frac{1}{||w||}$,

$$min_{r,w,b} = \frac{1}{2}||w||^2 \quad (15)$$

subject to

$$y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad (16)$$

This is called the optional margin classifier.

A linearly non-separable result is likely to be generated when there is an outlier in the dataset as this interrupts generating a clear boundary. We could apply L1 regularisation to deal with the outliers:

$$min_{r,w,b} = \frac{1}{2}||w||^2 + C\sum_{i=1}^{m} \xi_i \quad (17)$$

subject to

$$y^i(w^T x^{(i)} + b) >= 1 - \xi_i \quad (18)$$

$$\xi \geq 0 \quad (19)$$

A kernel trick is used when the hyperplane is not the linear decision boundary in the original dimension. The SVM kernel generates a linear boundary on a higher dimension. A Kernel function is:

$$K(x, z) = \phi(x)^T \phi(z) \quad (20)$$

where ϕ is feature mapping which maps from attributes to the features. By applying this, we could transform the data to a higher-dimensional space, find a linear decision boundary and project it to the original dimension. The kernel function enables us to find the optimal hyperplane without knowing explicit form of ϕ.

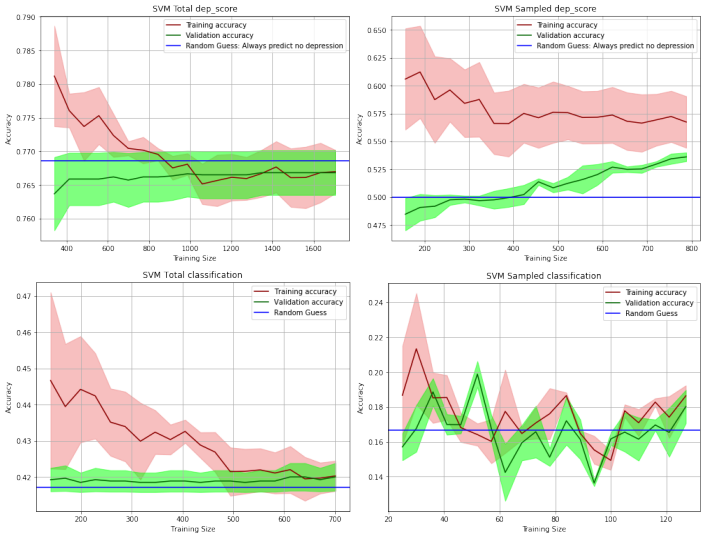

Fig. 7: Result in training SVM on the total data and the under-sampled data for classification (combined y targets) and `dep_score` (each y target). Data represented as mean and standard error (n=3).

2) Results: We can observe the training results of `dep_score` and classification in Fig. 7. It contains two curves: training and validation accuracy, and horizontal random guess prediction line (See equation 33). In this subsection, we will analyse each machine learning algorithm's results. We utilised total and sampled versions of `dep_score` and classification with four graphs. The result of the other y targets will be discussed in VI. FINDINGS.

For the top left plot (Total `dep_score`), the learning curve converges but the error (100 - accuracy) was high. Moreover, the validation curve is constant indicating that the learning was not efficient. It is likely that SVM is not a valid method or there could be no relationship between screen time and mental health.

In contrast, undersampled `dep_score` shows a much better learning curve (top right). It seems like the learning curve could converge if more data was available. The validation accuracy also exceeded a random guess and showed a relatively high F1 score (See Fig. 15). It is likely there is a relationship between mental health and screen time.

For total classification (combination of y), even though the curves converged, the error was very high and the learning curve's accuracy remained constant throughout the training (bottom left). This indicates that SVM is unable to detect the level of depression and anxiety. In addition, by looking at Fig. 16, the majority of the F1 scores are 0 which indicates that the algorithm does not have an ability to detect the *level* of depression.

By observing the bottom right plot, the learning curve highly fluctuates. This is probably because there is not enough data.

By observing the left plot of Fig. 8, it is clear that there is a relationship between the screen time and depression and anxiety. The learning curve converges with an accuracy higher than a random guess.

The right plot of Fig. 8, clearly shows that running classification is better than random guessing (0.167) although the accuracy is poor.

In terms of `dep_score`, it can be concluded that there is a relationship between the screen time and depression and anxiety. In terms of classification, although the total and under-sampled datasets show similar performance to a random guess, the over-sampled dataset shows better performance than a random guess.

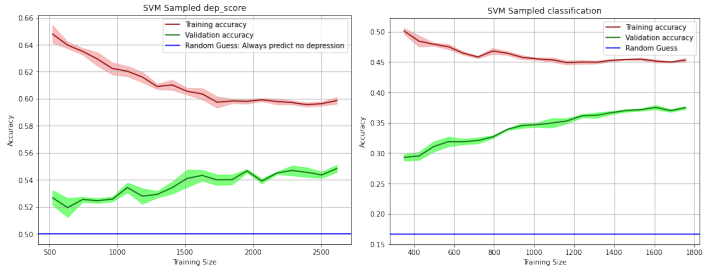

Fig. 8: Result in training SVM on the over-sampled data for `dep_score` (each y target) and classification (combined y targets). Data represented as mean and standard error (n=3).

B. Logistic Regression (LR)

LR is based on regression. However, it is different from linear regression since it works with categorical variables. It is called 'binary' if the outcome has two possible types, and 'multinomial' if the outcome has three or more possible types. The explanation of the algorithm below is for 'binary' LR. LR is an efficient model and easy to implement. It is also unlikely to overfit when there is a sufficient amount of data for training and when the data is low dimensional. However, logistic regression finds it hard to solve non-linear problems since it has a linear decision surface. Although we have shown that our data is not linearly separable into depression and no-depression, we thought that LR might provide some useful intuition.

1) Algorithm: Logistic regression is derived from linear regression: $y = \beta_0 + \beta_1 x$. The first step is to convert categorical variables into continuous variables. For this we use *odds* and *logit*. The *odds* implies the likelihood of the outcome. In other words, it denotes a 'successful outcome' over a 'failed outcome'. *logit* is the logarithm of the *odds* which enables categorical variables to be displayed in a continuous form. These can be expressed by the following equation:

$$odds : \frac{p(x)}{1-p(x)},\ logit : ln(\frac{p(x)}{1-p(x)}) \tag{21}$$

After applying logarithm function to *odds*, we can observe that $logit \in (-\infty, \infty)$. Remembering LR is based on the linear regression equation, $p(x)$ can be expressed like this:

$$ln(\frac{p(x)}{1-p(x)}) = \beta_0 + \beta_1 x,\ p(x) = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}} \tag{22}$$

The right hand side of the equation 22 is a sigmoid function, where $p(x) \in (0,1)$ and $x \in (-\infty, \infty)$ [20]. The following equation can be derived by applying a sample of n independent observations, $(x_i,\ y_i)$, $i = 1, 2, \cdots$ to the equation 22. Then, the probability of $y_i = 1$ is $p(x_i)$ and $y_i = 0$ is $1 - p(x_i)$.

$$p(x_i)^{y_i}[1-p(x_i)]^{1-y_i} \tag{23}$$

Finally, $log-likelihood$ which is the log form of the likelihood function, can be expressed by the following formula [20].

$$L(\beta) = \sum_{i=1}^{n} \{y_i ln[p(x_i)] + (1-y_i) ln[1-p(x_i)]\} \tag{24}$$

where $\beta = (\beta_0, \beta_1)$.
'Maximum likelihood estimation' is used for estimation when fitting a model.

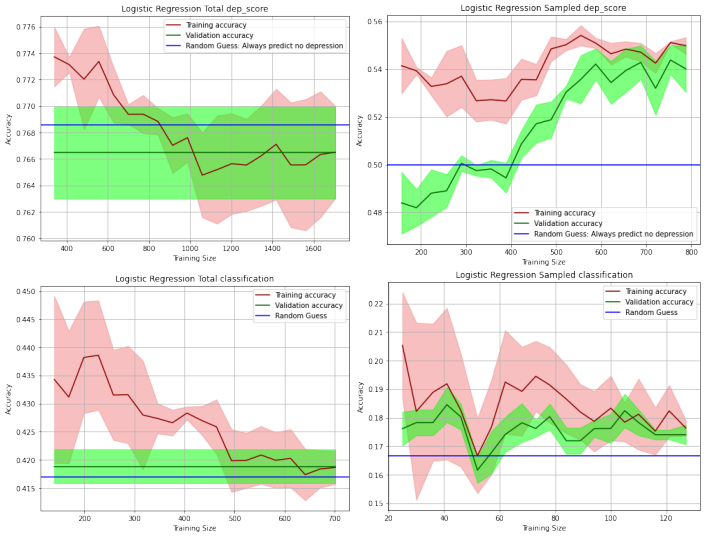

Fig. 9: Result in training Logistic Regression on the total data and the under-sampled data for classification (combined y targets) and `dep_score` (each y target). Data represented as mean and standard error (n=3).

2) Results: Fig. 9 shows results of the model. Firstly, for the total `dep_score` (top left), the training and the validation curve seem to converge as training size increases. The validation curve is lower than the random guess. This may imply that LR is not an appropriate model for predicting depression and anxiety or there could be no relationship between screen time and mental health. Also, the validation curve barely increases which means that the learning did not working well.

Secondly, with the under-sampled dataset for `dep_score` (top right), the learning curves look like they converge. It is worth noting that the validation accuracy is slightly higher than a random guess. This may indicate that more training data is likely to help.

For the total classification model (bottom left), the error is very high although the learning curve converges. The validation curve remains constant. These graphs indicate that LR model is not appropriate for detecting the level of depression and anxiety or there is no relationship between them.

For the sampled classification (bottom right), the curve is very unstable, also due to a lack of data.

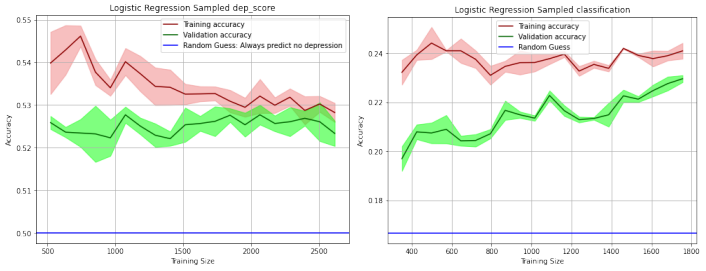

Fig. 10: Result in training Logistic Regression on the over-sampled data for `dep_score` (each y target) and classification (combined y targets). Data represented as mean and standard error (n=3).

By observing the left plot of Fig. 10, it is clear that there is a relationship between the screen time and depression and anxiety. The learning curve converges with accuracy higher than a random guess.

By looking the right plot of Fig. 10, it is clear that running classification is better than random guessing (0.167) although the accuracy is poor.

It seems like LR is able to detect `dep_score` from screen time with the under-sampled and over-sampled dataset. The performance of the classification is slightly better than random guessing.

C. KNN

k-Nearest Neighbours is a classification technique that outputs a class member for a new input based on a vote of its neighbours. The input is assigned to the class most common among the k nearest neighbours, where k is a predefined positive and typically small integer.

kNN is distribution free which means it does not have any assumption that the data is drawn from a given parametric family of probability distributions. It is a supervised machine learning algorithm. This algorithm works for both classification and regression tasks; but, we used only classification. To find the best value of k, we tried 10 possible values ranging from $k = 2$ to $k = 50$.

1) Algorithm: kNN iterates through each classified point and finds the distance to the new point. There are different ways to calculate this distance metric. For our purposes, we used **Euclidean distance** which would be represented by the following:

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \tag{25}$$

kNN then finds the k closest neighbours using the distance calculated and averages their class assignments to classify the new data point.

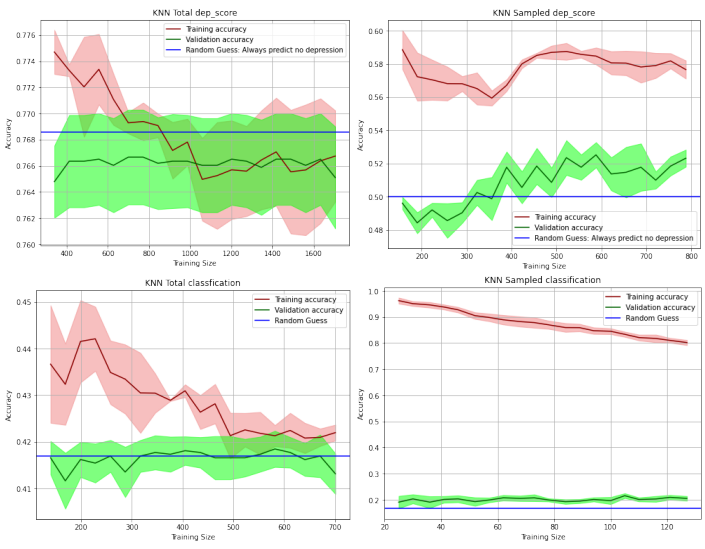

Fig. 11: Result in training KNN on the total data and the under-sampled data for classification (combined y targets) and `dep_score` (each y target). Data represented as mean and standard error (n=3).

2) Results: The result of applying kNN to the total and sampled dataset are represented in Fig. 11.

Running kNN on the total dataset for `dep_score` (top left) shows that the training and validation accuracy converges to a value lower than the accuracy of a random guess. When it was applied to the sampled dataset (top right), the result shows that as the training size increases, the validation accuracy increases. However, the value was still close to the accuracy of random guess. It seems like once again more data would help here.

When the algorithm was applied to the total classification dataset (bottom left), the result showed a similar trend to the top left figure.

Lastly, the learning curve seems to converge in bottom right figure. the validation curve slightly increases as the training size increases which is likely to indicate that the more data would be helpful.

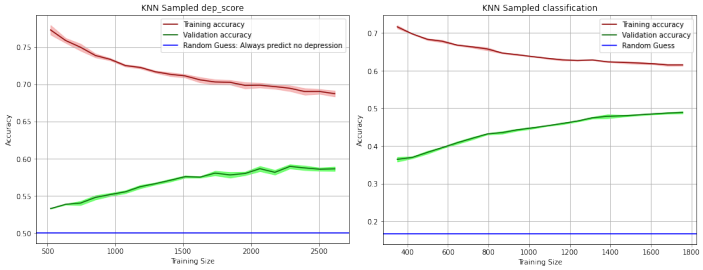

Fig. 12: Result in training KNN on the over-sampled data for `dep_score` (each y target) and classification (combined y targets). Data represented as mean and standard error (n=3).

Both graphs in Fig. 12 show that the learning curve converges. Their accuracy is better than random guessing. It would be possible to obtain better result if more data was available. KNN is able to predict depression and anxiety (`dep_score`) from the screen time. In addition, it shows much better performance than a random guess.

D. Random Forest (RF)

Random forest is a learning method for classification tasks that works by constructing a set of decision trees. Decision trees are easy to build and to interpret. A notable disadvantage of decision trees is a vulnerability to overfitting [16]. The RF algorithm is used to avoid this problem.

1) Algorithm: A Random forest is generated by incorporating multiple decision trees. Firstly, it samples the training dataset by bootstrapping. If bootstrapping is $true$, duplication is allowed; otherwise, it is not allowed.

Once the dataset is prepared by sampling, the model tries to do prediction. It is necessary to define the number of features (`max_features`) as the prediction could be varied depending on this value. The CART (Classification and Regression Trees) algorithm is applied to choose `max_features` and can be expressed like this:

$$Q_{left} = (x, y)|x_j \leq t_m \tag{26}$$

$$Q_{right}(\theta) = Q - Q_{left}(\theta) \tag{27}$$

$$G(Q, \theta) = \frac{n_{left}}{N_m} H(Q_{left}(\theta) + \frac{n_{right}}{N_m} H(Q_{right}(\theta) \tag{28}$$

$$\theta^* = \arg\min_{\theta} G(Q, \theta) \tag{29}$$

where Q represents data at node m. $Q_{left|right}$ is a partition and x_i is ith example. t_m, which is a threshold, segments data into two sets: $Q_{left}(\theta)$ for left hand side and $Q_{right}(\theta)$ for the right hand side.

The impurity function $H()$ which calculates $Gini$ and $Entropy$ can be expressed by the following formula [10].

$$p_{mk} = 1/N_m \sum_{y \in Q_m} I(y = k) \tag{30}$$

where k = 0 (indicating no depression), 1 (indicating depression) for node m. For instance, p_{m0} denotes the proportion of class 0 observation in node m.

$$Gini : H(Q_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{31}$$

$$Entropy : H(Q_m) = \sum_k p_{mk} log(p_{mk}) \tag{32}$$

Considering `min_sample_leaf` and `min_sample_split`, the above steps are repeated in order to select the suitable feature and threshold on each node until it reaches `max_depth`. This process generates a single decision tree. We need to repeat this process `n_estimator` time to build a forest. The final model aggregates the whole decision trees and this is why it is called Bagging as it combines 'bootstrapping' and 'aggregating'.

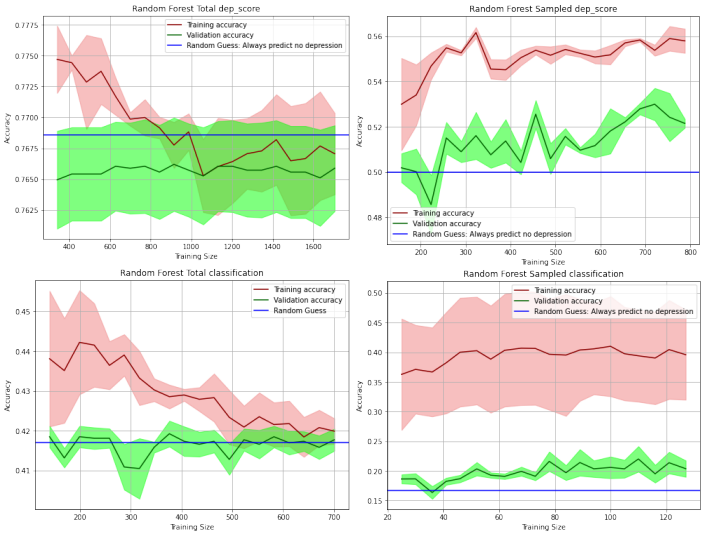

Fig. 13: Result in training RF on the total data and the under-sampled data for classification (combined y targets) and `dep_score` (each y target). Data represented as mean and standard error (n=3).

2) Results: You can see the result of applying random forest to our total and sampled datasets in Fig. 13.

The top left graph shows that the learning curve converges but the validation accuracy is lower than the random guess. Also, the validation curve is constant through out all training samples.

By observing the top right plot, the learning curve does not seem to converge and the validation curve highly fluctuates.

When we run RF on the total dataset with our combined y targets (bottom left), it can be observed that the result looks similar to the top left graph. The learning curve converges and the validation curve stays linear. This indicates that the learning was not very effective and RF might not be a suitable model to use.

The accuracy for both the training and validation is higher than a random guess. Both curves are fairly flat. This indicates that the learning was not very effective (bottom right).

All of these graphs imply that the random forest is unlikely to be a suitable model for predicting depression and anxiety with screen time. This is mainly due to the validation curve not ever significantly increasing.

Both graphs in Fig. 14 show that the learning curves begin to converge on each other. If there was more data available, we may observe more of a convergence. We could say that RF is able to predict mental health from the screen time. Although the

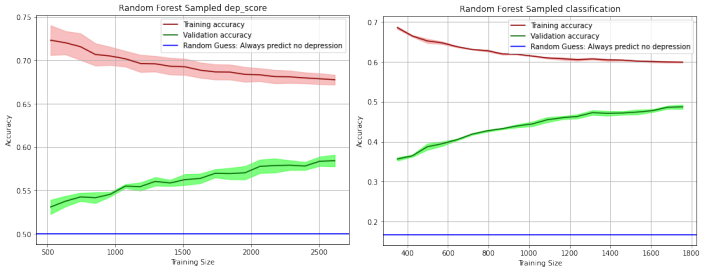

Fig. 14: Result in training RF on the over-sampled data for `dep_score` (each y target) and classification (combined y targets). Data represented as mean and standard error (n=3).

accuracies are higher than a random guess, they are still not ideal. The validation accuracies were around 50% to 60%.

VI. FINDINGS

This section presents all of the findings from the previous sections by conducting two experiments, *A* and *B*. The accuracy of a random guess is calculated by the following equation:

$$\text{Total (unbalanced) Random Guess} = \frac{max(Y s)}{\sum_{i=0}^{k} Y_i} \tag{33}$$

where $k = 1$ in experiment *A* and $k = 6$ in experiment *B*. Y_i indicates the number of data points in target Y_i.

A. Screen Time vs Each of Mental Health Related Target

Targets	ML Algo.	SVM	Logistic Reg	KNN	RF
	Metrics	Total / Under-Sampled / Over-sampled			
dep_score	F1: ND (0)	.87 / .58 / .54	.87 / .57 / .53	.87 / .53 / .61	.87 / .57 / .60
Random Guess Accuracy: .768 / .5 / .5	F1: D (1)	0 / .53 / .54	0 / .51 / .54	0 / .52 / .58	0 / .50 / .59
	Accuracy	.78 / .56 / .54	.78 / .54 / .53	.78 / .53 / .60	.78 / .54 / .60
dep_thoughts	F1: ND (0)	0 / .32 / .74	0 / .54 / .62	0 / .47 / .81	0 / .51 / .84
Random Guess Accuracy: .951 / .5 / .5	F1: D (1)	.97 / .53 / .70	.97 / .46 / .59	.97 / .47 / .77	.97 / .36 / .81
	Accuracy	.94 / .45 / .72	.94 / .50 / .61	.94 / .47 / .79	.94 / .45 / .83
has_dep_diag	F1: ND (0)	.96 / .48 / .65	.96 / .66 / .55	.96 / .52 / .71	.96 / .44 / .71
Random Guess Accuracy: .920 / .5 / .5	F1: D (1)	0 / .48 / .64	0 / 0 / .55	0 / .43 / .71	0 / .48 / .73
	Accuracy	.93 / .48 / .64	.93 / .49 / .55	.93 / .48 / .71	.93 / .46 / .72
prim_diag	F1: ND (0)	.89 / 0 / .56	.89 / 0 / .51	.89 / .58 / .63	.88 / .51 / .61
Random Guess Accuracy: .796 / .5 / .5	F1: D (1)	0 / .62 / .51	0 / .62 / .51	0 / .39 / .60	0 / .57 / .64
	Accuracy	.8 / .45 / .53	.8 / .45 / .51	.8 / .51 / .62	.79 / .54 / .62
secd_diag	F1: ND (0)	.94 / .56 / .63	.94 / .58 / .55	.94 / .50 / .69	.94 / .56 / .70
Random Guess Accuracy: .894 / .5 / .5	F1: D (1)	0 / .55 / .59	0 / .57 / .52	0 / .46 / .70	0 / .47 / .69
	Accuracy	.89 / .56 / .61	.89 / .57 / .53	.89 / .48 / .69	.89 / .52 / .69
panic score	F1: ND (0)	.99 / .62 / .73	.99 / 0 / .60	.99 / .56 / .83	.99 / .42 / .85
Random Guess Accuracy: .980 / .5 / .5	F1: D (1)	0 / .55 / .80	0 / .62 / .60	0 / .56 / .87	0 / .51 / .88
	Accuracy	.98 / .58 / .77	.98 / .44 / .60	.98 / .56 / .85	.98 / .47 / .87

Fig. 15: F1 score and accuracy of each mental health related variable.

- **Total**: For all 6 targets, each algorithm did not have the ability to detect depression or no depression (They all have an F1 score of 0 for either depression or no depression)
- **Under-sampled**: `dep_thoughts` and `has_dep_diag` have accuracies lower than a random guess (0.5). The accuracy of `dep_score` was slightly higher than a random guess. `prim_diag` also had accuracies higher than 0.5 apart from SVM and LR, RF showed the best accuracy among the 4 algorithms. For `secd_diag`, SVM and LR outperformed the other algorithms in both F1 score and accuracy. SVM and KNN outperform the other algorithms, showing a higher F1 score and accuracy. The best accuracy that we saw using our under-sampled dataset was 58% for `panic_score` using SVM.
- **Over-sampled**: We generally saw an increase in both the F1 score and accuracy for all algorithms and all target variables using the over-sampled dataset. We think that we could predict screen time from our target variables with our over-sampled dataset.

B. Screen Time vs Combination of Mental Health Related Targets

Target	ML Algo.	SVM	Logistic Reg	KNN	RF
	Metrics	Total / Under-Sampled / Over-sampled			
Combination of All Targets (Classification) Random Guess Accuracy: .417 / .167 / .167	F1: 1	.60 / .13 / .19	.60 / .11 / .21	.60 / .14 / .27	.60 / .17 / .26
	F1: 2	0 / .15 / .28	0 / .13 / .12	.02 / .10 / .41	0 / .07 / .37
	F1: 3	0 / 0 / .28	0 / .22 / .21	0 / .29 / .49	0 / .24 / .46
	F1: 4	0 / 0 / .46	0 / .15 / .24	0 / .21 / .63	0 / .15 / .64
	F1: 5	0 / .34 / .38	0 / .17 / .23	.03 / .26 / .5	0 / 0 / .49
	F1: 6	0 / 0 / .57	0 / .27 / .34	0 / .11 / .68	0 / 0 / .68
	Accuracy	.43 / .17 / .37	.43 / .17 / .24	.42 / .19 / .51	.43 / .12 / .50

Fig. 16: F1 score and accuracy of combined y targets

- **Total**: The F1 scores show that all of the algorithms do not work very well to classify the data. The accuracy is slightly higher than the random guess (0.417). However, we cannot rely on this accuracy because the F1 score is very poor.
- **Under-sampled**: All algorithms do not work well to classify the data by looking at the F1 scores. Although the accuracy of SVM, LR, and KNN is better than the random guess (0.167), we cannot rely on the value because again the F1 scores are very low.
- **Over-sampled**: It shows better performance than the total and under-sampled dataset. We found the accuracy is also much higher than a random guess but it still wasn't high enough for us to say that we could predict the level of depression from screen time.

C. Summary

By analysing experiment *A*, we found that the over-sampled dataset performs much better than the under-sampled dataset. We strongly believe this result is down to the fact that our over-sampling process allows the classifier to cheat. In other words, the model has already seen identical samples in the training set when testing. In reality, the test-set should be independent of the training set. However, it was not for our data.

To make a better model, we realised that we should first separate out the train and test data, and then perform random sampling only for the training data. Later, we also found that there are lots of sophisticated over-sampling algorithms such as SMOTE [5] and ADASYN [18]. These can be implemented using the `imblearn` package in Python. We also found out there are better under-sampling methods in this package such as `ClusterCentroids` and `NearMiss`. We think if we had more time and implemented these algorithms, we would see a much better result than our existing approach.

VII. CONCLUSION

It would not be an accurate representation to predict mental health from screen time using our over-sampled dataset. Therefore, this

conclusion is only based on the results from our total and under-sampled datasets.

Our findings show that it may be possible to predict whether someone has depression and anxiety or not, for each Y target (See `dep_score`, `prim_diag`, `secd_diag`, and `panic_score`) depending on the algorithm that you use. Some algorithms performed better on one target and worse on the others. However, we cannot solely rely on our algorithms because their accuracy is not by any means conclusive. The accuracies we saw were around 55% which are only slightly higher than a random guess (50%). We expect that for a specific y target, we would see an increase in accuracy if we had more data since the learning curves tended to converge (See top right plot of 7).

By analysing experiment *B*, we came to the conclusion that the machine learning algorithms we used cannot predict the level of depression and anxiety.

This result is not what we expected when we first started this project. We think this is mainly due to two main reasons.

A. Not enough data

The data provided to us was a very small dataset. It did not contain many variables that we could use to analyse the question. In addition, the variables that we could use had a large number of NaN values.

Training machine learning classifiers requires a large amount of data to generate a trustworthy model. Given the increased accuracy and converging curves, we saw when the training size increased, we believe that more data would strengthen these observations.

B. Out of date data

It is important to remember that thinking of what screen time means in the present day is a lot different to what it would have meant over a decade ago. This dataset is based off of data that was collected in 2008-2010. During this time, smartphone penetration was only between 22% and 36% [30] with even less among teenagers.

In addition to this, the main cause of depression and anxiety relating to screen time is believed to be heavily influenced by social media. A study in 2009 found that only 23% of teenagers accessed social media on their phones[33] and a 2018 study found that now over 95% of teenagers have access to a smartphone, 45% of which said they were 'online constantly'[34]. It is now believed that over 80% of teenagers use social media[31].

These statistics show the evolution in use of social media over the last decade which becomes increasingly relevant when you relate it back to our findings. The screen time variables that we used as inputs (`comp_wend`, `comp_week`, `text_week`, `text_wend`, `tv_week` and `tv_wend`) were not related to social media. Since social media is the main cause of depression relating to screen time vs tv or time spent on a computer [2], this might explain why we were not able to accurately predict depression and anxiety.

VIII. ADDITIONAL WORK

While our main aim was to find a relationship between columns *DS* and *DM*, we also focused on possible relationships between *DS* and *IS* to give a more complex analysis. For example, the column `tran_week`, indicates the average time a child spends per day in transportation on a typical weekday. During this time, they are more likely to use their phone if they own one, which would increase their screen time. If we could find a relation between 1. *IS* and *DS* and 2. *DS* and *DM*, then we could predict the relationship between *IS* and *DM*.

We observed that there might be a relationship between *IS* and *DS*. The gap between the random guess (25%)[1] and the prediction was a maximum of about 10%. By observing Fig. [1], it may be possible predict from *IS* to *DS* if more data was available. However, we have found that the majority of learning curves did not converge except for RF. Moreover, by observing the F1-score, we found that all the algorithms did not have the ability to differentiate between the number of hours in `text_week` and `text_wend`.

REFERENCES

[1] *Additional work*. https://git.io/JsitL. 2021.
[2] Elroy Boers et al. *Association of Screen Time and Depression in Adolescence*. https://doi.org/10.1001/jamapediatrics.2019.1759. 2019.
[3] Jason Brownlee. *Random Oversampling and Undersampling for Imbalanced Classification*. https://machinelearningmastery.com/random-oversampling-and-undersampling-for-imbalanced-classification/.
[4] Sunyool Chae. *Tutorial Kernel PCA*. https://www.youtube.com/watch?v=A30AFijdj4E. 2017.
[5] Nitesh V Chawla et al. "SMOTE: synthetic minority over-sampling technique". In: *Journal of artificial intelligence research* 16 (2002), pp. 321–357.
[6] *COMS30050: Screen Time Repo*. https://git.io/JsPBg. 2021.
[7] *Correlation Matrix and p-value: Additional Results*. https://git.io/JsiLW. 2021.
[8] *Data Modelling: Hyper-parameter search*. https://git.io/JsiLj. 2021.
[9] *Depression And Other Issues*. https://pulsetms.com/depression-issues/.
[10] Scikit-learn developers. *1.10. Decision Trees*. https://scikit-learn.org/stable/modules/tree.html. 2020.
[11] *Distribution of Targets*. https://git.io/JsiIB. 2021.
[12] Saurav K Dutta. *Statistical techniques for forensic accounting: Understanding the theory and application of data analysis*. FT Press, 2013.
[13] KI Ezukwoke and SJ Zareian. "KERNEL METHODS FOR PRINCIPAL COMPONENT ANALYSIS (PCA) A comparative study of classical and kernel pca." In: (2019).
[14] Karim F et al. *Social Media Use and Its Connection to Mental Health: A Systematic Review*. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7364393/#REF3. 2020.
[15] Suzi Gage. *B2551 - Assessing the association between screen time and anxiety in UK adolescents a prospective longitudinal study using the ALSPAC*. https://proposals.epi.bristol.ac.uk/?q=node/127766. 2016.
[16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media, 2009.
[17] Jan Hauke and Tomasz Kossowski. "Comparison of values of Pearson's and Spearman's correlation coefficient on the same sets of data". In: (2011).
[18] Haibo He et al. "ADASYN: Adaptive synthetic sampling approach for imbalanced learning". In: *2008 IEEE international joint conference on neural networks (IEEE world congress on computational intelligence)*. IEEE. 2008, pp. 1322–1328.
[19] Marti A. Hearst et al. "Support vector machines". In: *IEEE Intelligent Systems and their applications* 13.4 (1998), pp. 18–28.
[20] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*. Vol. 398. John Wiley & Sons, 2013.
[21] Ned H. Kalin. *The Critical Relationship Between Anxiety and Depression*. https://ajp.psychiatryonline.org/doi/10.1176/appi.ajp.2020.20030305. 2020.
[22] Divy Kangeyan. *t-SNE*. https://github.com/Divyagash/t-SNE/blob/master/tSNE_Presentation.pdf. 2017.
[23] Solomon Kullback and Richard A Leibler. "On information and sufficiency". In: *The annals of mathematical statistics* 22.1 (1951), pp. 79–86.
[24] Caroline Miller. *Does Social Media Cause Depression?* https://childmind.org/article/is-social-media-use-causing-depression/.
[25] Rita Osadchy. *Lecture: Kernel PCA*. 2011.
[26] *PCA with Kernels: Additional Results*. https://git.io/JsiLV. 2021.
[27] Elysia Richardson. *How Are Panic Disorders And Depression Related?* https://vistapineshealth.com/treatment/panic-disorders/and-depression/.
[28] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. "Kernel principal component analysis". In: *International conference on artificial neural networks*. Springer. 1997, pp. 583–588.
[29] Samuel Sanford Shapiro and Martin B Wilk. "An analysis of variance test for normality (complete samples)". In: *Biometrika* 52.3/4 (1965), pp. 591–611.
[30] *Smartphone penetration in the United States from 2008 to 2014*. https://www.statista.com/statistics/218529/us-martphone-penetration-since-2008/. 2012.
[31] *Social Media use in 2021*. https://www.pewresearch.org/internet/2021/04/07/social-media-use-in-2021/. 2021.
[32] *t-SNE: Additional Results*. https://git.io/JsiLN. 2021.
[33] *Teens and mobile phones*. https://www.pewresearch.org/internet/2010/04/20/teens-and-mobile-phones-3/. 2010.
[34] *Teens, Social media & Technology*. https://www.pewresearch.org/internet/2018/05/31/teens-social-media-technology-2018/. 2018.
[35] Laurens Van der Maaten and Geoffrey Hinton. *Visualizing data using t-SNE*. 2008.
[36] Pablo Vandenabeele. *What's the difference between anxiety and depression?* https://www.bupa.co.uk/newsroom/ourviews/2017/10/anxiety-depression. 2021.

[1] It is 25% because there are 4 categories in `text_week` and `text_wend`: Not at all, Less than 1 hour, 1-2 hours, and 3 or more hours.

Reflective Discussion

Junoh Park (ov18075)

In the ADS group project, I think that my contribution was good. The main question of our project is "Does the screen use at age 16 affect the level of depression and anxiety at age 18?". To answer the question, understanding, and handling the dataset was important. Therefore, all our group members contributed on the categorising the data into various categories and targets. I also contributed on it. Then, to explore data, we started from the linear regression. I built the model of linear regressions and ran the model with categorised dataset. To run the model, the data wrangling step is required, since some data we chosen were not numerical. It was the form of the answers in questionaries like 'Not at all', 'Less than 1 hour', '1-2 hours' and '3 or more hours'. Then, I suggested to convert them into the number form such as 'Not at all' to 0, 'Less than 1 hour' to 1 and so on. And also, I converted them by my hand. After converting the dataset, I ran the linear regression models with various combinations of categories to get some insight. After that, I and our group members decided which machine learning model to use. Then, since some group members finished the data wrangling and others built frame of the machine learning code, I was the person who merged the data and model, applied the hyperparameter search on the dataset and changed the parameter on the machine learning method. I and some members worked in this part together. In writing the report, I have attended on writing the machine learning part. So, I introduced the models and gave background information. Also, I wrote the result part too.

Basically, we arranged the meeting twice a week. It was Monday and Thursday. I attended most of the meeting and shared idea with our group member at least twice a week. Even on the day without meeting like Wednesday, I contact with our members personally and shared the results constantly.

One of the strengths of our model I think is that the result is based on the various machine learning models. Since every machine learning model has own strengths and weaknesses, the result could be biased if few models are used in the project. However, since we used various methods not only machine learning but also finding relations, I believe that it could be a strength. Another strength of our model is categorising. We selected not only directly related variables, but also indirectly related variables. And we had process to categorise them and use in the model. I also believe that approach with various possibilities open, and not just thinking about direct relations is a strength.

However, there is also a weakness. Since we have suffered from nan value, we chose under-sampling and over-sampling methods. However, since we proceeded random sampling for over-sampling before separate the data into train and test set, it could bring some biased result. And it will be better if we use over-sampling algorithm.

The project is massively related with the data wrangling part in the unit. As taught in the lectures, data wrangling is one of the most important part to handle the data also one of the parts that we should focus.

In this project, I believe that I have learnt a lot. Before this project, data science was not very serious problem for me since I believe that it is not tricky. However, with the project, I found that handling and analysing the data requires a lot of efforts then expected. Also, when I made a little problem on data or code, I found that the results change a lot. So I have also learnt that since small change could bring tremendous consequence, I have to be careful on working with dataset. It was great experience to me and I feel that I can do better job on data science in future.