Using Machine Learning Methods to predict whether or not somebody has a Mental Health issue within the Technology Industry

1. Introduction

Mental health is becoming a more prevalent issue in society annually, with one in four people being affected in any given year [1]. A mental health issue is defined as 'a health condition involving changes in emotion, thinking or behaviours (or a combination of these)' [2]. Identifying mental health issues within employees is beneficial for all parties. For the employer, it would allow them to offer treatment for mental illnesses sooner, which would reduce the amount of time that an employee's productivity would be reduced [3]. For the employee, it would likely improve their happiness, physical health and extend their life expectancy [4].

This project focuses on whether a person working in the technology field can be diagnosed with a mental health issue by answering 63 questions from a 2016 survey performed by Open Sourcing Mental Illness (OSMI) [5].The questions range from whether the person is self-employed, their age, some of their opinions on mental illness and whether there is any family history of a mental disorder.

1.3 Aim

The aim of this project is to use machine learning methods to predict whether a respondent to the OSMI survey has a mental illness, in addition to finding which features are the most important to the result of predictions. The final model should have an accuracy of at least 75%, as it is important to have reliable results as a misdiagnosis could potentially be life threatening. This should mean that a person is able to get help quickly, and efficiently, being beneficial to both employer and employee. However, it is important to note that these machine learning models should not be used exclusively to diagnose somebody with a mental illness and a professional should be consulted before a final decision on whether somebody does/does not have a mental health disorder.

2. Literature Review

A 2020 study by Ashley E. Tate, Ryan C. McCabe, Henrik Larsson, Sebastian Lundström, Paul Lichtenstein and Ralf Kuja-Halkola provided a very strong starting point for this project. Their paper, titled 'Predicting mental health problems in adolescence using machine learning techniques' looked at predicting mental health problems in adolescence using machine learning [6]. Within their paper they looked at a variety of machine learning techniques, including Random Forrest classification, Support Vector Machines, Neural Networks and XGBoost. Their research found that the two most affective models for their application was Random Forrest Classification, and Support Vector Machines, this is shown below in Figure 1. Once the models had been tuned, the Random Forest Classifier was found to be the most accurate, although the difference between the Random Forrest Classifier and

the Support Vector Machine was minimal. Therefore, it was worth exploring both of the classifiers to see which one would be more accurate for this application.

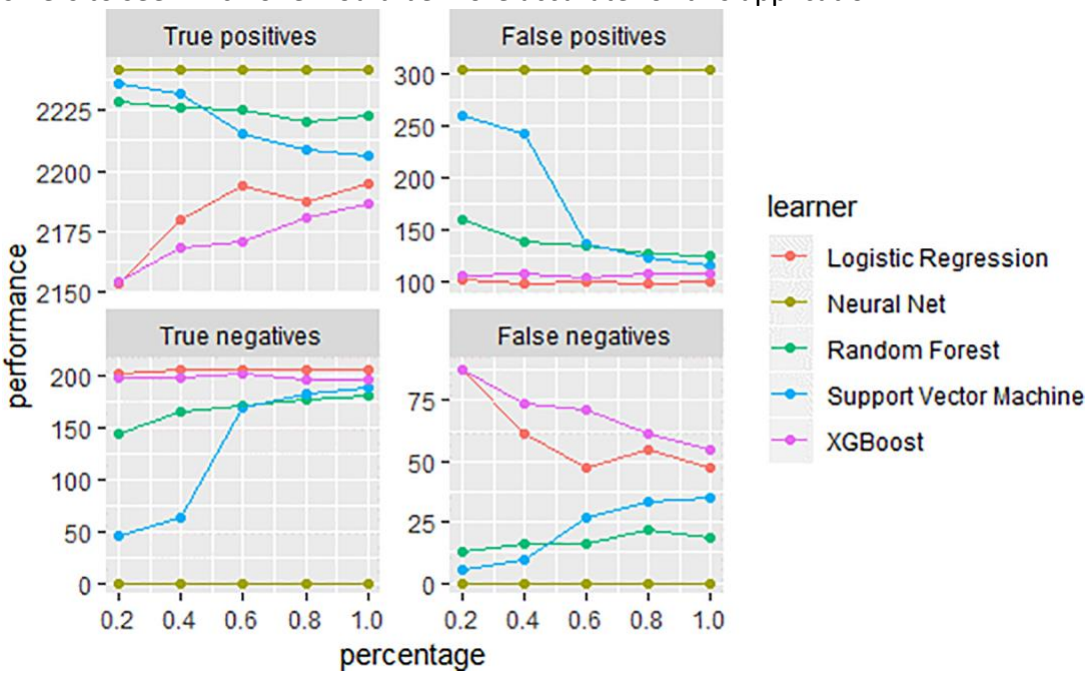

Figure 1 – Showing the performance of various machine learning techniques. [6]

Another paper by Chang Su, Zhenxing Xu, Jyotishman Pathak and Fei Wang titled 'Deep learning in mental health outcome research: a scoping review' [7] provides an insight into using a deep learning method to predict whether somebody has a mental health disorder. However, their research involves looking at many different sources to build a profile on the individual which gives them a vast dataset. Therefore, it was not strictly relevant to the research in this paper as there was not enough datapoints to build an effective neural network.

3. Background Information

3.2 Linear Support Vector Classification

The support vector machine (SVM) is a linear model for classification and regression problems. The basic principle of the support vector machine is that they map pattern vectors to a high-dimensional feature space where a 'best' separating hyperplane is constructed [8]. As two sets of data can be divided in infinitely many ways by a hyperplane, the SVM tries to find the best possible position for the hyper plane.

Take the following linear discriminant function:

$$g(x) = w^T x + w_0 \tag{1}$$

The decision rule would be the following:

$$w^T x + w_0 \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow \begin{cases} \omega_1 \; with \; corresponding \; numeric \; value, y_i = +1 \\ \omega_2 \; with \; corresponding \; numeric \; value, y_i = -1 \end{cases} \tag{2}$$

Therefore, the training points are considered to be correctly classified if:

$$y_i(w^T x_i + w_0) > 0 \; for \; all \; i \tag{3}$$

Figure 2a attempts to visualise this, with Figure 2b a showing how infinite hyperplanes can intersect the data. Figure 2b is also showing how the distance of the 'best' hyperplane is the plane which sum of the distances from the separating hyperplanes closest to each class is the largest (line labelled A). It can be assumed that the larger the sum of the distances separating the hyperplanes, the better the generalisation error.

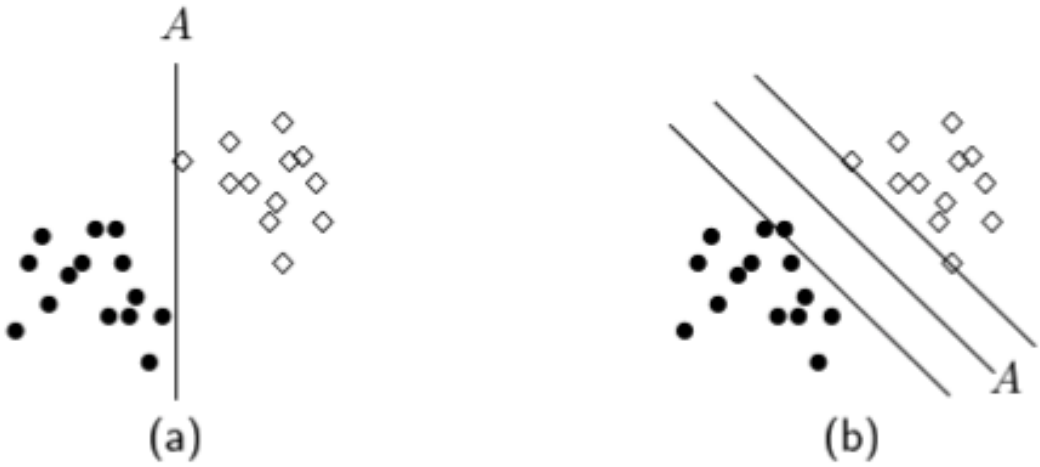

Figure 2 – Two linearly separatable sets of data with separating hyperplane, labelled A. The two lines flanking hyperplane A in figure b leaves the closest points at maximum distance. [8]

3.3 Decision Trees

Decision trees take a similar approach to a human playing a game of *Guess Who*. The decision tree will perform a set of tests on the values of descriptive features and use the answers to the tests to try to formulate a prediction. The key advantage of decision trees the ease of understanding of them. They follow a very linear form which is easily understandable, working top down, and splitting according to parameters defined within the tree. Decision trees would be extremely useful in this scenario as there are many variables which can be split easily, to give a prediction for whether the individual currently has a mental health disease.

The decision tree is formed from a series of nodes and branches. The 'starting node' is called the root node, each node is then connected by branches, with interior nodes and leaf nodes, which are the 'terminating' nodes. Each interior node (a node which is not a leaf node) defines a test to be performed on a descriptive feature. Each leaf node represents a predicted level of the target feature [9].

The dataset is split into two parts, a training set and a testing set. The training set is used to train the model to predict the outcome, and the testing set is used to test the accuracy and performance of the model.

When a decision tree makes a prediction, it starts by testing the value of the descriptive feature at the root node of the tree. Once this result has been found, the tree can then decide which of the root node's children to proceed to. This process is repeated down the tree, until a leaf node is reached, and the value is said to have been predicted.

For a decision tree to be most effective, it must be as shallow as possible, being sure to tests the informative features early on within the tree. This is done by making the root node which reduces the entropy of the model as much as possible. Entropy is related to the probability of an outcome, when there is a high probability of an event, there will be a low

entropy. Conversely, if there is a low probability of an event, the entropy would be high. To calculate the entropy for a model, a weighted sum of the logs of the probabilities for each outcome is computed. The formula for entropy is:

$$H(t) = -\sum_{i=1}^{I} (P(t = i) \times \log_s(P(t = i))) \quad (4)$$

Information Gain is another common metric which is used, which makes use of entropy. Information Gain is found using the formula:

$$IG(d, \mathcal{D}) = H(t, \mathcal{D}) - rem(d, \mathcal{D}) \quad (5)$$

Where $H(d, \mathcal{D})$ is:

$$H(t, \mathcal{D}) = -\sum_{I \in levels(t)} (P(t = I) \times \log_2(P(t = I))) \quad (6)$$

And $rem(d, \mathcal{D})$ is:

$$rem(d, \mathcal{D}) = \sum_{I \in levels(d)} \frac{|\mathcal{D}_{d=I}|}{|\mathcal{D}|} \times H(t, \mathcal{D}_{d=I}) \quad (7)$$

Where $\frac{|\mathcal{D}_{d=I}|}{|\mathcal{D}|}$ is the weighting and $H(t, \mathcal{D}_{d=I})$ is the entropy of the partition $\mathcal{D}_{d=I}$.

Where the information gain is largest, it is considered to be the best 'question' to be put into the node to ensure the largest reduction in entropy and minimise the depth of the decision tree.

By making depth of a decision tree too high, there is a greater chance of overfitting the data, and by making the decision tree too shallow there is a greater chance of underfitting the data. It is therefore important to consider the stop conditions within a decision tree. This is done in two ways, limiting the tree size, and by tree pruning. The way in which limiting the tree size works is quite obvious, where the user defines variables such as, minimum number of splits for a node split, defining the maximum number of terminal nodes and the maximum depth of the tree. Tree pruning is a technique which reduces the size of the tree by removing sections of the tree, reducing complexity of the final classifier and improving accuracy and reducing overfitting [10]. Tree pruning is split into two methods, pre and post pruning. In pre-pruning, splitting is stopped if splitting the current node does not improve the entropy by a pre-set value. In post-pruning, a full decision tree is made, then the tree is pruned after calculating the cross-validation accuracy at each level [10].

An example decision tree, and its corresponding data set, is shown below in Figure 3. The Figure represents a game of *Guess Who*.

Man	Long Hair	Glasses	Name
Yes	No	Yes	Brian
Yes	No	No	John
No	Yes	No	Aphra
No	No	No	Aoife

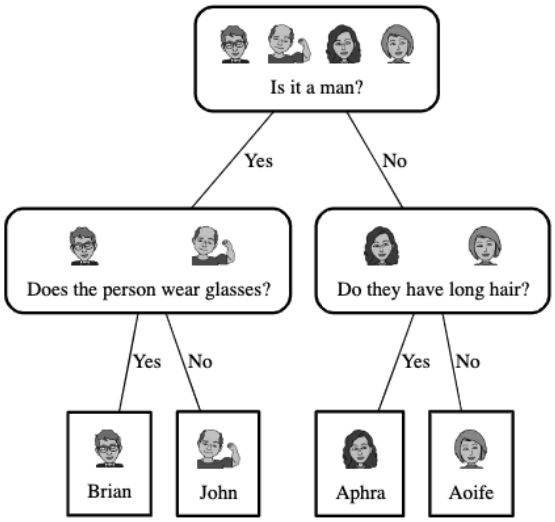

Figure 3 – An example dataset and decision tree. [9]

3.4 Random Forests

The simplest way of thinking about the Random Forest method is to simply consider multiple Decision Trees (discussed in Section 3.3) which are merged together to get a more accurate and more stable prediction [11]. The trees within a Random Forest work together to form an ensemble that combines the predictions made by decision trees using a majority vote decision rule [8].

The largest advantage which Random Forests has over ordinary Decision Trees is that Random Forests are less likely to overfit the data. This is because in a Random Forest, multiple Decision Trees are combined, which predicts better than just a single tree. Multiple mediocre models which are combined are more accurate than a single 'good' model.

Each tree within the Random Forest contains a random selection of the training data. As addition trees are added the resolution of the feature space is increased leading to much higher accuracy in classification. Figure 4 demonstrates this.

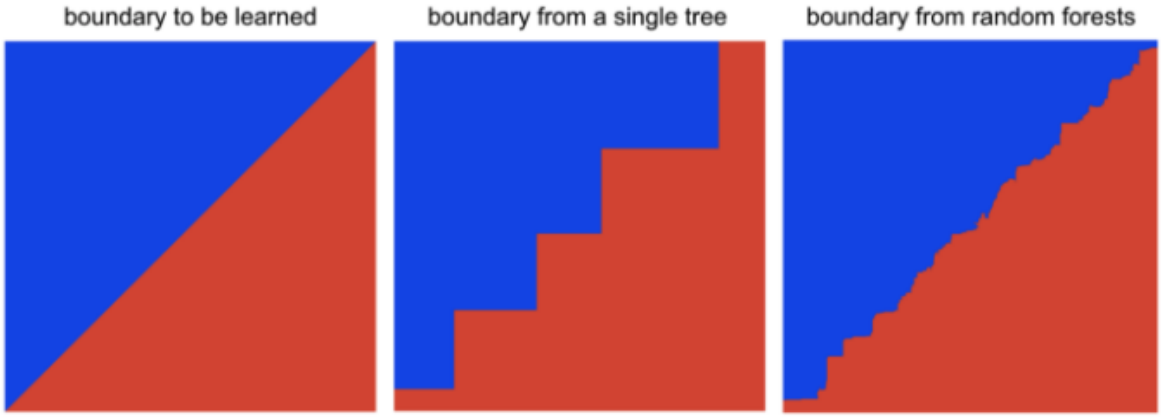

Figure 4 – A figure to demonstrate the higher resolution which is given by using Random Forests. [12]

The most important thing for a high performing model is to have very low correlation between the individual trees [13]. By doing this, any errors between the trees will be evened out and the correct results should be predicted.

Random Forests have shown classification accuracies comparable with the best current classifiers on many datasets [8]. This makes Random Forests a very attractive method to be used for any classification problem.

3.4.1 Bootstrap Method

Each tree is constructed using a bootstrap sample of the data. By using the bootstrap method, roughly two thirds of the sample are used to train the model, whilst one third is used for testing the model. If there are n patterns in the dataset, n samples are taken, with replacement, to generate a bootstrap set of size n [8]. By using this random selection of data to train the trees, it creates trees which are not correlated to each other, preventing overfitting of the model. Due to the sampling technique used, we do not need ridiculous amounts of data to train the model.

3.4.2 Feature Randomness

When using a decision tree, while splitting a node, all features within the data are considered and the feature which leads to the highest information gain is chosen. However, when using Random Forests, each tree can only pick features from its random subset of features [13]. This allows for an increased variation between the trees in the model, lowering correlation across the trees which in turn increases diversification.

4. Method

4.1 Data

The data which was used throughout the experiment was sourced from the Open Source Mental Illness (OSMI) Mental Health in Tech Survey, which was performed in 2016. The data is open-source and is available from the OSMI website, or from the website Kaggle.com. The survey was completed by 1433 participants with the 'aim to measure attitudes towards mental health in the tech workplace,' [14] . Despite more recent surveys being conducted by OSMI, the annual survey from 2016 had the most responses, making it more useable when performing machine learning methods. In the 2016 report, there is a wide range of data, collating information about the respondent's workplace (e.g. location/number of staff), the stance of the business regarding mental health (whether they offer mental health benefits etc.), the respondent's view point on mental health (whether they would feel discussing it with their employer), as well as information regarding whether the respondent feels as though they have any mental health issues/have been diagnosed with a mental health illness. There was a total of 62 questions asked to each respondent.

All pre-processing and machine learning were used using the SciKit Learn package, within a JupyterLab notebook running Python 3.

4.2 Pre-processing

Upon first finding the data, it was evident that there would have to be some form of pre-processing performed on the data, due to frequent missing entries and many answers to the survey which were the same answer in different form. A good example of this is when looking at the 'gender' question. Within the question asking the respondent for their gender, just for the answer 'male' there were many variations, including 'm', 'M', 'Male', 'male' and even 'I'm a man why didn't you make this a drop down question. You should of asked sex? And I would of answered yes please. Seriously how much text can this take?'.

4.2.1 Cleaning the Data

The first task to tidy up the data, was to change the column titles. They were changed to be descriptive of the feature, whilst being more concise which made them easier to work with.

Following the column changes, changes to the aforementioned variations within the 'gender' column were made. Genders were split into three distinct categories. There was a total of 70 unique values for the question 'What is your Gender', so it is clear that needed to be cleaned up before any sort of analysis could be conducted. The data was split into the following three categories:

1. Male

This was to cover anybody who had referred to themselves as a cisgender/transgender male, or anything which could be used to identify them as a man.

2. Female

This was to cover anybody who had referred to themselves as a cisgender/transgender male, or anything which could be used to identify them as a woman.

3. Other

This was to cover anybody who had identified as non-binary/gender fluid, anybody who did not elect to reveal their gender or any gender which could not be quantified.

The genders were converted so that the Male = 1, Female = 2, Other = 3.

When considering age, there was a vast range of answers which were given, with a maximum value of 323, and a minimum value of 3. For the answers which were outside of the working age range (15– 64, defined by the Organisation for Economic Co-operation and Development [15]) was set to be the mean of all of the values of ages, as these responses were assumed to be false.

Finally, when cleaning the data, the location of the respondent's work was removed as there was negligible difference between their living location and working location. In addition to working location being dropped, both why/why not questions were dropped. This was because the reason why they would choose to share any mental health/physical health issues within an interview was not important, only that the individual would or would not.

4.2.2 Missing Data

The next step was to deal with missing values, of which there were plenty. The strategy which was used to deal with these missing values, was to drop questions which had less than a 50% result rate, as any interpolated data would not be accurate enough to use. When the response rate was over 50%, the missing values were assigned the most commonly occurring value in the column. This was done using SimpleImputer from sci-kit learn. The columns which asked 'why or why not' were also dropped, as they would be too difficult to analyse and get any meaningful information and the only information which was wanted was whether or not the person would share that information in the first place.

Prior to imputing any data, the data was separated into two categories, the United States of America, and the rest of the world. This was done so that the rest of the world was not given a state, which was done to prevent any skewing of results due to a more commonly occurring state, from the most frequent method used when imputing data. Once the data had been imputed, the two data sets were then recombined.

4.2.3 Encoding Data

To help the machine learning methods to learn from the data, the data was encoded. This means that instead of giving the methods 'yes', 'no' or any other text, they receive a number. This had already been done for the gender column but needed to be done for many other columns.

This was performed manually for all answers which could be a yes/no/maybe question to ensure the maximum amount of control over the results could be obtained.

To encode the categorical variables, location of user, get_dummies() was used from Sci-Kit Learn. At this point, states were dropped from the data set, as there were less than 30 individual data points for each state, and therefore reliable conclusions could not be drawn. It suggested that you have an absolute minimum of 30 entries for a sample [16].

There was a wide range of values which were returned when the individuals were asked to describe their work position. These values were split into multiple columns which were to roughly describe what the individual did for work. Once the values were split into multiple columns, the data could then be encoded.

4.2.4 Other Adjustments

The survey was completed by respondents in 53 different countries, with the majority of them being from either the United States of America, the United Kingdom, Canada, Germany, the Netherlands and Australia. Just these countries account for 86% of all of the data points. Therefore, all countries with fewer than 30 individual datapoints, were grouped together under the umbrella of 'Other'. The columns which represent where people work, and which state that they work in, were also dropped due to very little difference between their living location and working location.

The survey was actually split into two sections, depending on the answer to the initial question, which was given to the respondent, which asked them if they were self-employed, since questions regarding their employer's mental health resources were not applicable. Therefore, the data frame was split into the two groups, of self-employed, and not self-employed. Where the respondents were self-employed, all columns which

corresponded to their current employers' stance on mental health were dropped, as they were all not a number. This allowed for tests to be performed on both sets individually as well as both tests being able to be performed together.

4.3 Machine Learning

4.3.1 Choosing a machine learning algorithm

As discussed within the project aims, the aim of the project was to accurately predict whether an individual currently has a mental health disorder. An algorithm cheat sheet which was provided by scikit-learn was used to help to choose the best algorithm for the application. The cheat sheet which was used can be shown below in Figure 5. From the cheat sheet, it can be seen that Linear Support Vector Classifier (Linear SVC) is a good place to start as an algorithm, as well as the information from the paper formerly discussed [6]. Another model which was considered was the Random Forest algorithm. Random Forests were considered due to their high accuracy when used in classification models, which was also considered due to information in the paper formerly discussed [6].

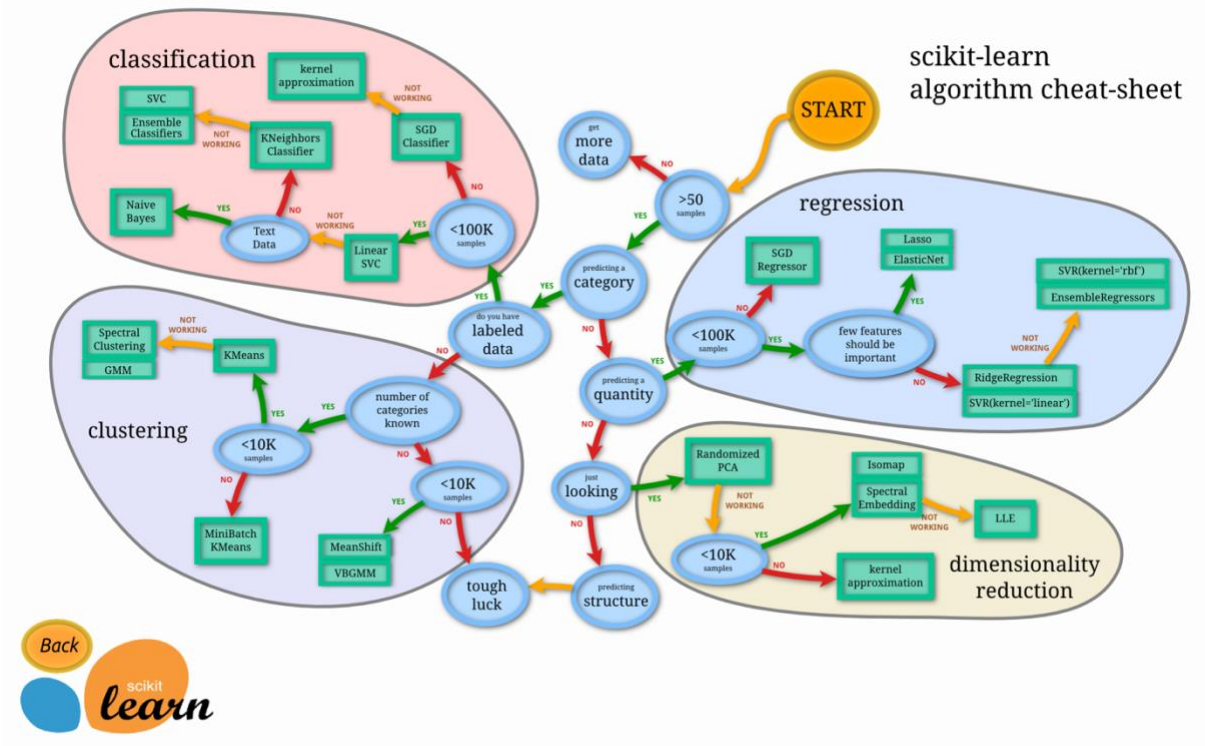

Figure 5 – scikit-learn algorithm cheat-sheet [17]

Models were created for both the Random Forests classifier and the Linear Support Vector classifier so that the accuracy of both models could be compared, before selecting the model which gives the greater accuracy to be used for subsequent tests and analysis.

4.3 Improving the Model

The model was initially run with a set of typical values, for the Random Forest classifier, n_estimators was set to 1000, and max_depth to 10. For the Linear SVC, the tolerance for stopping criteria was set to 1e-5.

The model was then improved and run again to maximise the accuracy and true positive rate of the model. The parameters were improved by using RandomizedSearchCV and GridSearchCV. The final parameters are included within the results section of the report.

Results and Discussion

5.1 Initial Results

Figure 6 contains the accuracy and confusion matrix from the initial test. It can be seen that for both the employed and self-employed data, the Random Forest is significantly more accurate than that for the Linear Support Vector model. As the Random Forest gave a much higher accuracy score, it was decided that only the Random Forest would be refined to improve results. It is also worth noting that when the initial tests were run, convergence warnings were returned for both of the self-employed data sets. This is because there are too few datapoints within the self-employed data set for there the model to converge. The warnings were then suppressed. It is also worth noting that the confusion matrix is 3x3 as the answer to the question originally was yes/no/maybe.

```
==== Employed Random Forest ====
 Accuracy:  0.7616279069767442
 [[110  20   9]
 [  2 125   8]
 [ 14  29  27]]

==== Self Employed Random Forest ====
 Accuracy:  0.7241379310344828
 [[21  3  3]
 [ 2 31  3]
 [ 4  9 11]]

==== Employed Linear Support Vector ====
 Accuracy:  0.49127906976744184
 [[69 11 59]
 [10 49 76]
 [ 7 12 51]]

==== Self Employed Linear Suppoert Vector ====
 Accuracy:  0.47126436781609193
 [[22  1  4]
 [14 11 11]
 [13  3  8]]
```

Figure 6 – Initial results

5.2 Parameter Tuning

RandomizedSearchCV was used to increase the accuracy of the Random Forest model by changing the method's parameters. The accuracy results are shown below in Figure 7. The comparison between the accuracy of the RandomizedSearchCV Model and the base model is shown in Figure 8.

```
==== Employed RandomizedSearchCV parameter model ====
 Accuracy:  0.7587209302325582

==== Self-Employed RandomizedSearchCV parameter model ====
 Accuracy:  0.7126436781609196
```

Figure 7 – Accuracy results after using RandomizedSearchCV

```
==== Employed RandomizedSearchCV against Base Model change ====
 -0.3816793893129778 %

==== Self Employed RandomizedSearchCV against Base Model change ====
 -1.5873015873015854 %
```

Figure 8 – Comparison of the accuracy between the RandomizedSearchCV model against the Base Model

Following the use of RandomizedSearchCV, GridSearchCV was used to try to further refine the parameters of the Random Forest model. The accuracy results are shown below in Figure 9. The comparison of the accuracy between the GridSearchCV model and the Base Model and the RandomizedSearchCV model is shown in Figure 10.

```
==== Employed GridSearchCV model ====
 Accuracy:  0.7616279069767442

==== Self Employed GridSearchCV model ====
 Accuracy:  0.735632183908046
```

Figure 9 – Accuracy results after using GridSearchCV

```
=========================================
 ==== GRIDSEARCHCV AGAINST BASE MODEL ====
 =========================================

==== Employed GridSearchCV against Base Model change ====
 0.0 %

==== Self Employed GridSearchCV against Base Model change ====
 1.5873015873015854 %

=========================================
 ==== GRIDSEARCHCV AGAINST RANDOMIZEDSEARCHCV MODEL ====
 =========================================

==== Employed GridSearchCV against RandomizedSearchCV change ====
 0.38314176245210796 %

==== Self Employed GridSearchCV against RandomizedSearchCV change ====
 3.225806451612899 %
```

Figure 10 – Comparison of the accuracy between GridSearchCV model and the Base Model and RandomizedSearchCV model

GridSearchCV has the highest accuracy for both models, compared to both the RandomizedCV model and the Base Model. Therefore, for the final models, the parameters which were found using GridSearchCV. The parameters which were found by GridSearchCV and were used in the final model are shown below in Figure 11.

```
==== Employed GridSearchCV Parameters ====
 {'bootstrap': True, 'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 8, 'n_estimators': 1000}

==== Self Employed GridSearchCV Paramets ====
 {'bootstrap': True, 'max_depth': 45, 'max_features': 'sqrt', 'min_samples_leaf': 3, 'min_samples_split': 12, 'n_estimators': 300}
```

Figure 11 – Parameters used within the final model

5.3 The Final Model

The final results of both Random Forest models are shown below in Figure 12.

```
==== Employed Random Forest ====
 Accuracy:  0.752906976744186
 [[109  22   8]
 [  4 123   8]
 [ 15  28  27]]

==== Self Employed Random Forest ====
 Accuracy:  0.7241379310344828
 [[21  3  3]
 [ 2 33  1]
 [ 6  9  9]]
```

Figure 12 – Final accuracy and confusion matrices for both the employed and self-employed Random Forest models

The confusion matrices for the final models are shown below in Figure 13 and Figure 14. As previously discussed, the confusion matrix is 3x3 as the answer to the initial question was yes/no/maybe. The important values are the diagonal, as those are the results which are the correct predictions, the True Positives/True Negatives/True Maybes (the labels which have been correctly predicted).

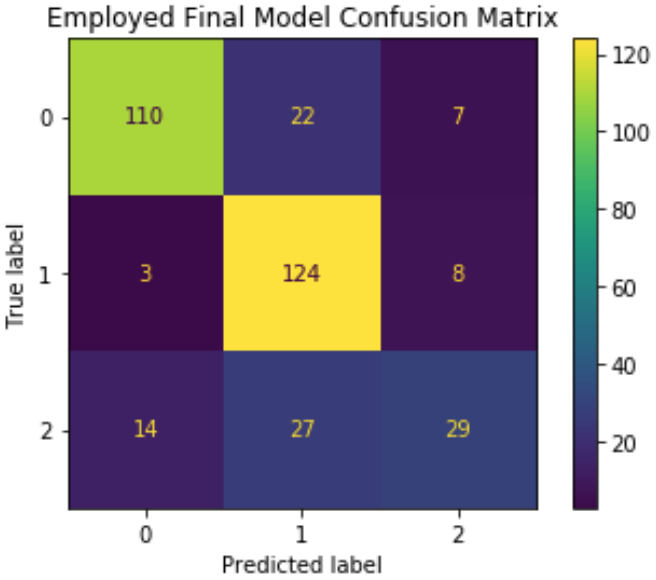

Figure 13 – Employed Final Model Confusion Matrix

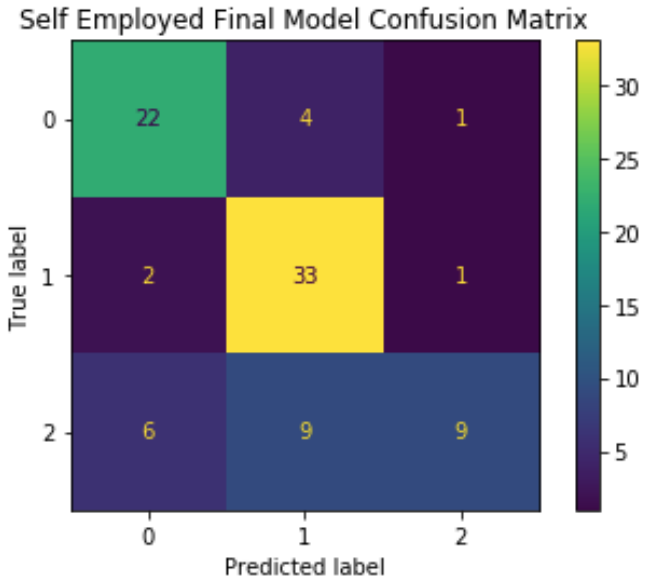

Figure 14 – Self Employed Final Model Confusion Matrix

5.3 Feature Importance

The final aim of the project was to find the features for both the employed and self-employed which are the most important to determining the outcome of the prediction. Within Sci-Kit Learn it is possible to determine the feature importance with the .feature_importances_. The results of the most important features are shown below in Figure 15 and Figure 16. Only the top 10 features are shown, as they are by far the most significant, with the rest of the features being combined into the umbrella of 'other'.

The most important feature for the employed model is shown to be 'If you have a mental health issue, do you feel that it interferes with your work when NOT being treated effectively?' followed closely by 'Have you had a mental health issue in the past'. The most important feature was surprising here and would warrant more investigation in the future.

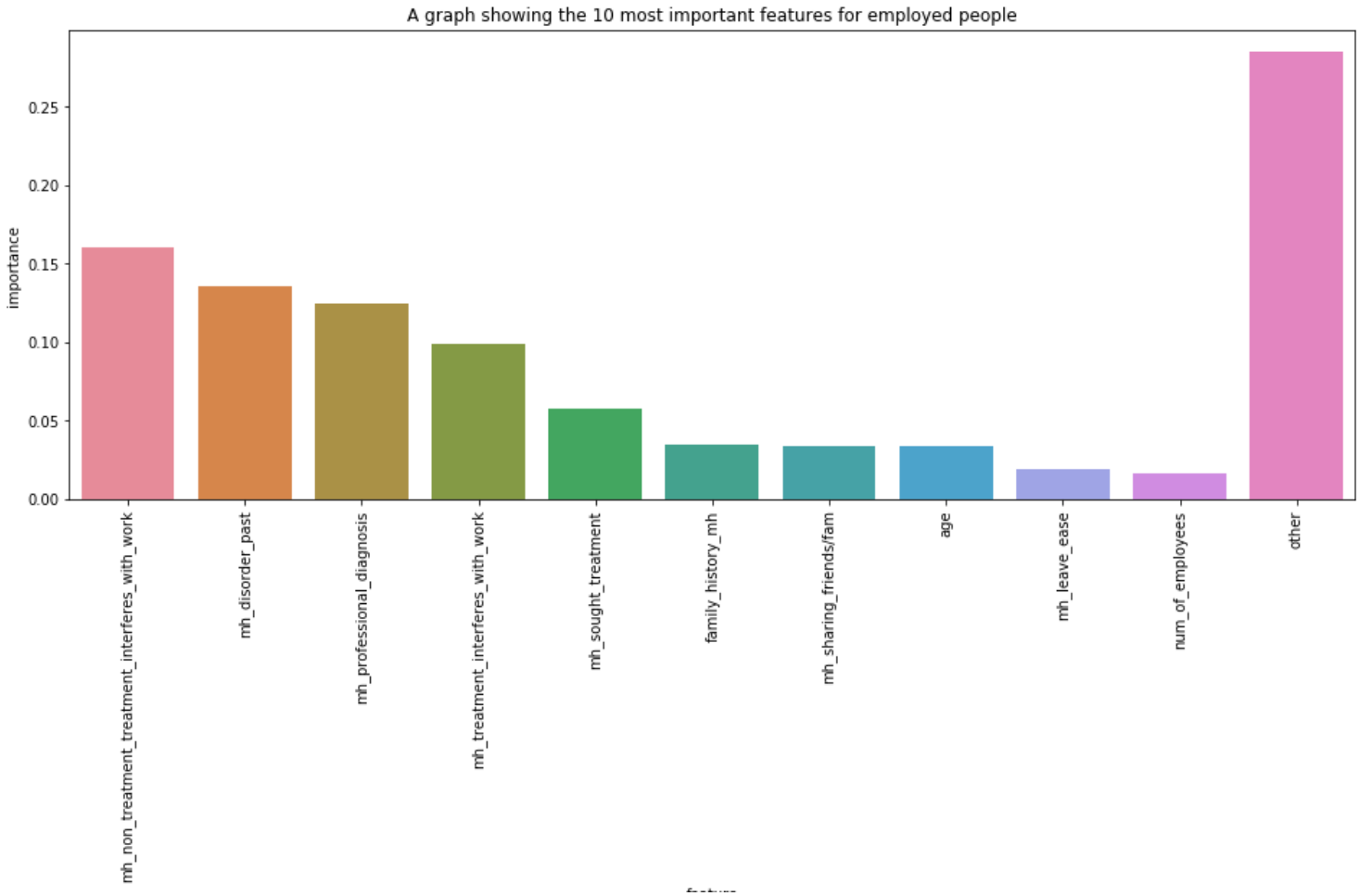

Figure 15 – Graph showing 10 most important features for employed people

The self-employed most important feature was found to be the history of a previous mental health disorder, which is something which would be expected.

The results show that a large proportion of the most important features are to do with work, such as how any treatment might interfere with work.

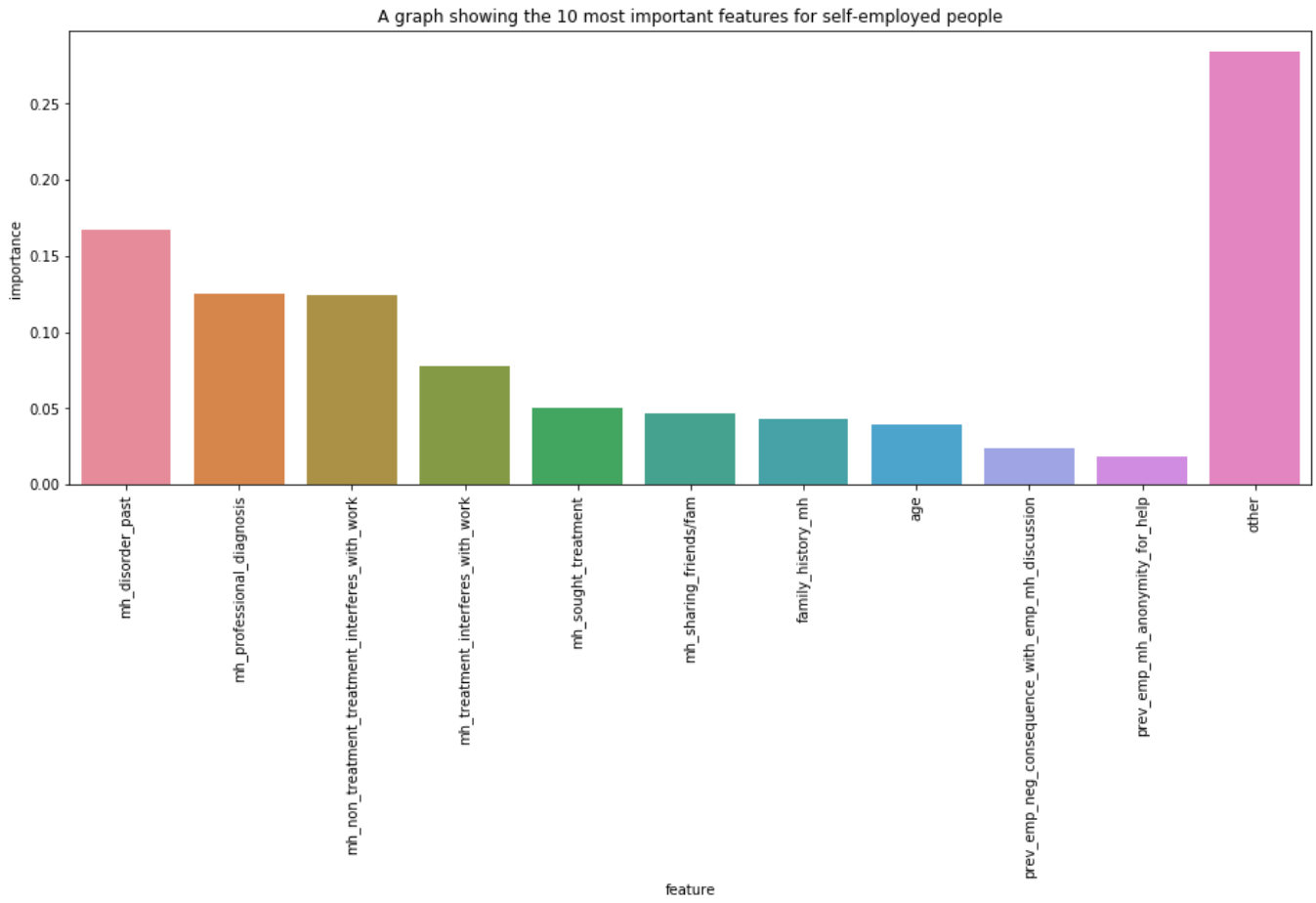

Figure 16 – Graph showing the 10 most important features for self-employed people

6. Conclusion

With a final accuracy for the employed model of 75.3% and 72.4% for the self-employed model, the models could be described as adequate, but they were not high enough accuracy to be used to reliably to diagnose mental health disorders. This is due to the severity of the repercussions if somebody is mis-diagnosed. However, it is accurate enough to be able to be used as a tool to help somebody in the medical field to diagnose a mental health disorder, or as an in-work survey to try to direct help before it might be needed.

The most important features were also found for both employed and self-employed correspondents, which were found to be the same three for both, being whether or not they had a mental health disorder previously, whether they have had a professional diagnosis and whether they have not had treatment because they believe that it would interfere with work.

Overall, the report contains a strong basis for using these models to predict whether or not a respondent has a mental health disorder, which can be built on with further work to make the results more accurate and reduce the number of questions which the respondent has to answer. It also provides key information for employers within the technology field of how it is important to support their employees to maximise their effectiveness as an employer.

6.2 Further Work

I think a natural progression for this work could be to look at other surveys done by the OSMI and compare the results from those surveys with the ones within this paper. Another way to extend this work would be to look at whether the survey could be reduced in length and have the same accuracy of results, as people are more likely to complete a survey if they have to answer fewer questions.

7. References

[1] mind, "Mental health problems – an introduction," mind, 1 January 2017. [Online]. Available: https://www.mind.org.uk/information-support/types-of-mental-health-problems/mental-health-problems-introduction/about-mental-health-problems/#:~:text=Mental%20health%20problems%20affect%20around,as%20schizophrenia%20and%20bipolar%20disorder.. [Accessed 9 January 2020].

[2] R. Parekh, "What is Mental Illness?," American Psychiatric Association, 1 August 2018. [Online]. Available: https://www.psychiatry.org/patients-families/what-is-mental-illness#:~:text=Mental%20illnesses%20are%20health%20conditions,Mental%20illness%20is%20common.. [Accessed 9 January 2020].

[3] A. L. C. L. I. S. J. U. R. E. G. M. V. M. R. W. A. Beck, "Severity of Depression and Magnitude of Productivity Loss," *Annals of Family Medicine,* vol. 9, no. 4, pp. 305-311, 2011.

[4] E. S. M. H. M. K. J. M. S. G. D. B. T. Russ, "Association between psychological distress and mortality: individual participant pooled analysis of 10 prospective cohort studies," *BMJ,* vol. 345, no. 1, pp. 1-14, 2012.

[5] OSMI, *OSMI Mental Health in Tech Survey 2016,* Indianapolis: OSMI, 2016.

[6] R. M. H. L. S. P. L. R. K.-H. A. Tate, " Predicting mental health problems in adolescence using machine learning techniques," *PLOS ONE,* vol. 15, no. 4, p. e0230389, 2020.

[7] Z. X. J. P. F. W. C. Su, "Deep learning in mental health outcome research: a scoping review," *Translational Psychiatry,* vol. 10, no. 1, 2020.

[8] C. K. C. G. Webb A, Statistical Pattern Recognition, London: John Wiley & Sons, Incorporated, 2011.

[9] N. B. D. A. Kelleher J, Fundamentals of machine learning for predictive data analytics : algorithms, worked examples, and case studies, Cambridge, Massachusetts: Electronic reproduction, 1974.

[10] Y. A, "Decision Trees," towards data science, 11 January 2019. [Online]. Available: https://towardsdatascience.com/decision-trees-d07e0f420175#:~:text=Tree%20prunning%20can%20be%20done,some%20preset(Threshold)%20values.. [Accessed 15 November 2020].

[11] D. N, "A complete guide to the random forest algorithm," Builtin, 3 September 2020. [Online]. Available: https://builtin.com/data-science/random-forest-algorithm. [Accessed 15 November 2020].

[1 D. H, “Why random forests outperform decision trees,” towards data science, 28
2] October 2018. [Online]. Available: https://towardsdatascience.com/why-random-forests-outperform-decision-trees-1b0f175a0b5. [Accessed 15 November 2020].

[1 Y. T., “Understanding Random Forest,” towards data science, 12 June 2019. [Online].
3] Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2. [Accessed 15 November 2020].

[1 Open Sourcing Mental Illness Ltd, “Reserach,” OSMI, 13 November 2020. [Online].
4] Available: https://osmihelp.org/research. [Accessed 13 November 2020].

[1 OECD, “Working age population,” OECD, 31 December 2018. [Online]. Available:
5] https://data.oecd.org/pop/working-age-population.htm. [Accessed 13 November 2020].

[1 Statistic Solutions, “Sample Size Formula,” Statistics Solutions, 1 January 2020. [Online].
6] Available: https://www.statisticssolutions.com/sample-size-formula/. [Accessed 14 November 2020].

[1 Scikit-learn, “Choosing the right estimator,” Scikit-learn developers, 1 August 2020.
7] [Online]. Available: https://scikit-learn.org/stable/tutorial/machine_learning_map/index.html. [Accessed 14 November 2020].