Prediction of Depression Risks from International Personality Item Pool

Cláudio Filipe P. Gomes - *gomes@student.dei.uc.pt*
Maria Inês A. Roseiro - *miroseiro@student.dei.uc.pt*
Faculdade de Ciências e Tecnologia
Universidade de Coimbra
Pólo II - Pinhal de Marrocos
3030-290 Coimbra

Keywords
Big Five Model Five Factor Model Personality IPIP Depression Malaise Inventory Supervised Learning Regression Models Machine Learning

I. Introduction

Personality is widely modelled by psychologists into a Five-Factor Model (FFM) that can predict an individual's behaviour, motivation, and interaction with the surrounding environment. Furthermore, previous studies suggest a link between FFM and mental or behavioural disorders, such as depression[1]. The aforementioned disorder is affecting an ever-growing number of people, showing signs of sadness, loss of interest, low self-esteem, poor concentration, among other symptoms[2].

This study aims to demonstrate that it is possible to diagnose someone's depression symptoms by using their responses to questions from the International Personality Item Pool (IPIP), based on the Five Factor Model. Moreover, these predictions can be used to detect other psychological illnesses.

Over the last decades, a group of studies has demonstrated that personality traits are formal stable in adulthood, which means that there are no aged related shifts and individuals maintain very similar traits coefficients. Based on this information, our study relies on individuals at the age of 50, because it leads us to more solid information, with important similarities between normal and personality disorder[3].

Finally, we compared several supervised learning models regarding their detection performance.

A. Concepts

Before delving further into the paper, we explore the background concepts of Personality, Depression and Supervised Learning Algorithms.

1) Personality: The definition of personality is not consensual, however, the core goal of personality psychology is the prediction and explanation of traits and behaviours based on motivation and interactions with one's environment. Commonly, personality is divided into five statistically identified personality traits, called the Five-Factor Model (FFM), which is regarded by many as the most extensively used and comprehensive model. Furthermore, the traits are: openness to experience, conscientiousness, extraversion, agreeableness, and neuroticism (McCrae & Costa, 2013). [4] For our study, we used a data set that uses 50 questions from the International Personality Item Pool (IPIP), which is a public domain collection of items for use in personality tests[5].

2) Depression: According to the *World Health Organization*, depression is a highly prevalent mental disorder that is estimated to affect 322 million people in 2017. Generally, depressed individuals are sad, have no interest or pleasure, feel guilty or low self-worth, have disturbed sleep or appetite, feel tired, and have poor concentration. Moreover, depressive disorders have symptoms that range in terms of their severity and duration, which can, in some cases, be long-lasting and recurrent, preventing the individual to function at school or work, and causing difficulty in coping with his daily life. In the most severe cases, depression can lead to suicide.[2] The data set used in this paper uses 9 of the Malaise Inventory items, a set of self-completion questions which combine to measure levels of psychological distress, or depression.[6]

3) Supervised Learning Algorithms: Supervised Learning Algorithms are tasks in Machine Learning that map inputs into outputs, based on example input-output pairs. Usually, the input is a set composed by vectors of features and the output is a target vector. Any row from the set of features would have its corresponding target value. In the present paper, we used five Supervised Learning Models: linear regression, gradient boosting; random forest; k-nearest neighbours algorithm and support-vector machine.

4) Decision trees: In statistics and machine learning, decision trees are predictive models that go from observations about an item to conclusions about the item's target value (as in, they go along the branches until they find a leaf). When the target variable is discrete, they are called classification trees, whereas when the target variable is continuous they are called regression trees. Any given conclusion can be explained just by tracking the condition in the tree, which is one of the major advantages of those predictive models. Supervised learning models using decision trees "come closest to meeting the requirements for serving as an off-the-shelf procedure for data mining", say Hastie et al., "because it is invariant under scaling and various other transformations of feature values, is robust to inclusion of irrelevant features, and produces inspectable

models. However, they are seldom accurate".[7]

5) Ensemble: Ensemble methods are techniques that join multiple learning algorithms into a single predictive model with better performance than any of the constituent algorithms alone. Usually, ensemble methods are distinguished into two families: averaging methods, which uses several independent algorithms and averages their predictions; and boosting methods, which sequentially uses algorithms, each one reducing the bias of the combined predictive model.

B. State-of-the-art

Empirical studies of personality and psychopathology suggest that many mental disorders are linked to abnormal variations in personality traits frameworks, such as FFM (Krueger & Eaton, 2010). These studies indicate the existence of an informative marker between traits of personality and depression. The research made by Kotov et al. (2010) demonstrates that patients with depression scored higher on Neuroticism, and lower on Extraversion and Conscientiousness, with large differences and no significant distinctions on the Openness and Agreeableness scales. There aforementioned relation between the interaction of the three particular traits and patients with depression has been well-replicated in experiences[8].

II. Data set

We used data from the National Child Development Study: Sweep 8, 2008-2009, which works on a set of 17 638 individuals born in Great Britain during one week in March 1958.[9]

At age 50, the participants answered 50 items from IPIP, that correlate to FFM, with 10 items per factor rated on a 5-point rating scale (the Cronbach alpha reliabilities were 0,78 for openness to experience, 0,77 for conscientiousness, 0,87 for extraversion, 0,81 for agreeableness, and 0,88 for neuroticism). Furthermore, the participants answered a 9-item Malaise inventory, with a Cronbach alpha reliability of 0,79.[10]

A. Data selection

Fuelled by the goal of detecting depression risks using an individual's IPIP responses, we selected the participant's sex and responses to the 50 IPIP questions as features for our supervised learning models. As a target vector, we chose their Malaise Inventory score, which was designed as estimation of depression risk[6].

B. Data preprocessing

The selected set was then verified for missing values and outliers. No values were missing and some values were deleted with the following conservative definition of an extreme outlier.

Definition II.1. Given a value x, the first quartile $Q1$, the third quartile $Q3$, and the interquartile $IQR = Q3 - Q1$, x is said to be an extreme outlier if $x < Q1 - 3 \cdot IQR$ or $x > Q3 + 3 \cdot IQR$.

After the removal, as the sex feature is in different units compared to the remaining features, we normalised the features to a 0-1 range, in order to not affect the supervised learning models.

Finally, the data set was split into a training set and a testing set, with a proportion of 80% to 20%, and the data set can be considered ready.

III. Methodology

After defining the problem statement, we are going to present the performance unit we chose for the problem, along with a baseline performance we perceive as a reasonable mark supervised learning models should surpass.

A. Problem Statement

Given an IPIP response from an individual, a supervised learning system must predict their Malaise Inventory score, which correlates with their depression risk[6].

B. Mean Absolute Error and Mean Squared Error

We measure a given system's performance by computing their Mean Absolute Error (MAE) in a provided set, commonly a testing set.

Definition III.1. Given two n-sized sets X and Y, the Mean Absolute Error MAE is such that

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=0}^{n} X_i - Y_i \tag{1}$$

However, sometimes, we reference the Mean Squared Error (MSE).

Definition III.2. Given two n-sized sets X and Y, the Mean Squared Error MSE is such that

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=0}^{n} (X_i - Y_i)^2 \tag{2}$$

C. Baseline Performance

An initial baseline performance was established before applying supervised learning models. If none of the models can beat this mark, then our problem may not be suited for machine learning and would need a different approach. We guessed the depression risk using the median of the target vector, which returned a MAE of 1,2715 when applied to the testing set. Afterwards, we fit, train, and test the supervised learning models, whose design, hyper-parameters, and results are presented in the following sections.

IV. Models

Five supervised learning models were implemented in this work: linear regression, gradient boosting; random forest; k-nearest neighbours algorithm and support-vector machine. We will present how each model works and what are their hyper-parameters.

1) Linear regression: Linear regression is one of the simplest regression methods. A linear relationship is assumed between our target, a dependent variable y, and a set of independent features $x = (x_1, ..., x_r)$, where where r is the number of predictors.

This method calculates the estimators of the regression coefficients (the predicted weights). The coefficients define the estimated regression function $f(x) = b_0 + b_1x_1 + b_rx_r$, which captures dependencies between our features and target.

Definition IV.1. Having x as the input training data and y as the data labels, the Hypothesis function for Linear Regression is such that

$$y = \theta_1 + \theta_1.x \tag{3}$$

with θ_1 as the intercept value and θ_2 as the coefficient of x.

Once the best values found, the best fit line is given. Therefore, when the model is adopted to prediction, it will predict the value of y for the input value of x.

2) Gradient boosting: Gradient boosting is a technique which produces an ensemble of weak prediction models, in our case decision trees. More specifically, we used Gradient Boosted Regression Trees (GBRT), which consider additive predictive models of the form:

$$F(x) = \sum_{m=1}^{M} \gamma_m h_m(x) \tag{4}$$

where $h_m(x)$ are decision trees. GBRT then expands the additive model in a greedy fashion:

$$F_m(x) = F_{m-1}(x) + \gamma_m h_m(x), \tag{5}$$

where h_m tries to minimize the loss L, given the previous ensemble F_{m-1}:

$$h_m = \arg\min_h \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) + h(x_i)). \tag{6}$$

In our case, the initial model F_0 is the mean of target values when the loss function is least squares regression, or a quantile when using other loss functions. GBRT attempts to solve this minimisation problem with the method of steepest descent, where the steepest descent direction is the negative gradient of the loss function evaluated at the current model F_{m-1}, which can be calculated for any differentiable loss function:

$$F_m(x) = F_{m-1}(x) - \gamma_m \sum_{i=1}^{n} \nabla_F L(y_i, F_{m-1}(x_i)) \tag{7}$$

where the step length γ_m is chosen using line search:

$$\gamma_m = \arg\min_\gamma \sum_{i=1}^{n} L(y_i, F_{m-1}(x_i) - \gamma \frac{\partial L(y_i, F_{m-1}(x_i))}{\partial F_{m-1}(x_i)}) \tag{8}$$

The following hyper-parameters were tuned:

Loss function We can choose one of the following: least squares regression, least absolute deviation, and the combination of both;

Number of estimators Number of decision trees used in the additive model;

Max depth Maximum depth of each decision tree;

Number of samples to split Minimum number of samples required to split an internal node;

Number of samples to leaf Minimum number of samples required to be at a leaf node;

Max number of features Number of features to consider when looking for the best split.

Usually, gradient boosting techniques are well suited to a wide range of problem. The decision trees make GBRT a versatile method that can be used for data that has features with different units or has features that can't be easily normalised, where some features are very small and other features are very large.[11]

3) Random forest: Random forest (RF) predictive models are another ensemble based technique for classification and regression problems. Although very deep decision trees can learn highly irregular patterns, they suffer from over-fitting, with high variance. Taking this into account, random forest models try to reduce the variance by averaging multiple decision trees, trained on different splits of the same training set. RF model averages a set of M decision trees:

$$F(x) = \frac{\sum_{m=1}^{M} h_m(x)}{M} \tag{9}$$

where $h_m(x)$ is a modified decision tree that chooses a random subset of the features.

The following hyper-parameters were tuned:

Criterion Function to measure the quality of a split. We can choose MAE or MSE;

Number of estimators Number of decision trees used in the additive model;

Max depth Maximum depth of each decision tree;

Number of samples to split Minimum number of samples required to split an internal node;

Number of samples to leaf Minimum number of samples required to be at a leaf node;

Max number of features Number of features to consider when looking for the best split.

Like gradient boosting, RF is a versatile method due to the use of decision trees.[11]

4) K-nearest neighbours: K-Nearest Neighbour algorithm is a simple predictive model that stores all the available cases and classifies a new one based on a similarity measure. Essentially, it summarises information from the K most similar instances to a given observation to perform a majority vote. The similarity is defined applying a distance metric between two data points. In our case, Manhattan Distance was employed.

Definition IV.2. For two given points P and Q, Manhattan Distance $d(P, Q)$ is such that

$$d(P, Q) = \sqrt{\sum (q_i - p_i)^2} \tag{10}$$

Neighbours-based regression is adopted when the data labels are continuous, where the assigned label to a query

point is computed based on the labels means of its neighbours. This can be done in two different methods, the *K Nearest Neighbour Regression* or *Radius Nearest Neighbour Regression*. In our case,the *K Nearest Neighbour Regression* method was applied. The following hyper-parameters were tuned:

Number of neighbours Number of neighbours to be used by the point queries;

Leaf Size Leaf size passed to the algorithms;

P Parameter for the *Minkowski* metric, which performs distance calculations.

Weights approach The neighbourhood contribute to the regression. The choice is made between uniform weight, where query point value is calculated with a simple majority from the nearest neighbours, or distance based weight, which assigns weights proportional to the inverse of the distance from the query point.

Algorithm The algorithm used to compute the nearest neighbours. The options are: *brute-force*, that as the name suggests, brute-forces the distances between all pairs of points in the data set,*K-d tree*, who tries to reduce the required number of distance calculations by encoding aggregate distance information for the sample, and *ball tree*, which was meant to improve *K-d tree*, by dividing recursively the data into nodes defined by a centroid and a radius, such that each point in the node lies within the hyper-sphere defined by this 2 parameters.

5) Support-vector machine: Support-vector machines (SVM) are supervised learning algorithms based on the idea of finding a hyper-plane that best divides a data set into two distinct classes, as the following figure shows:

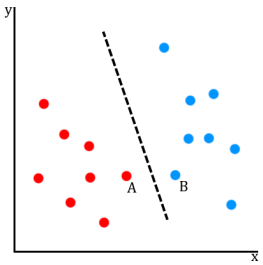

Fig. 1: Hyper-plane illustration

where vectors A and B are called support-vectors, because they are the nearest points of the line (hyper-plane) that separates both red and blue classes. If the support-vectors were removed, the line equation would be altered. However, there are cases such that no line can possible separate the data set in two classes, for example:

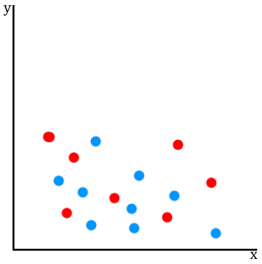

Fig. 2: Scattered point vectors with no clear hyper-plane

This is were hyper-planes come into play. By using an approach called ”kernel trick”, which maps the data set features into higher dimensions, we can separate the classes with hyper-planes (sub-spaces whose dimension is one less than that of its ambient space) and solve the classification problem:

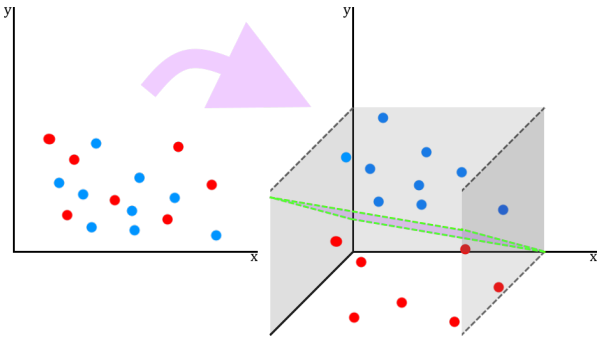

Fig. 3: Kernel trick

This technique can also be transformed to a regression problem, which is our case.

The following hyper-parameters were tuned:

Kernel Kernel function used. We can choose linear, polynomial, radial basis, or sigmoid functions;

C Regularisation parameter, to be decreased if the data set is too noisy;

ε It specifies the ε-tube within which no penalty is associated in the training loss function with points predicted within a distance ε from the actual value. We can imagine this tube as a zone around the hyper-plane where errors inside aren't penalised as those outside.

SVM can be very accurate models, but the higher the hyper-planes' dimensions, the higher will be their generalisation errors. Furthermore, due to their training time, they aren't very suited to large data sets and work better on smaller and cleaner sets.

V. Training and Testing

Fortunately, our models, with the initial hyper-parameters, surpassed the baseline reference, with linear regression having a MAE of 1,0543, gradient boosting having a MAE of 0,9433; random forest having a MAE of 1,0076; k-nearest neighbours algorithm having a MAE of 0,9861 and support-vector machine having a MAE of 0,9810.

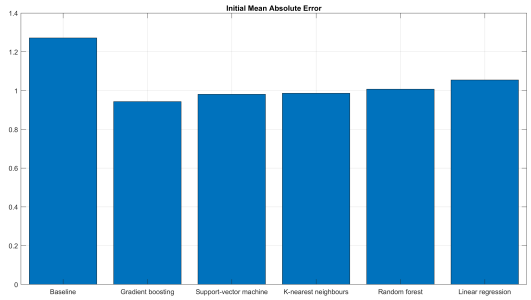

Fig. 4: Initial Mean Absolute Error

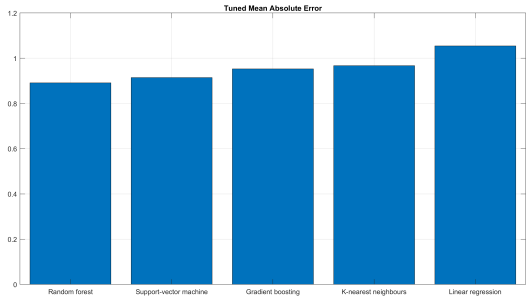

Fig. 5: Tuned Mean Absolute Error

Those initial results open the path to further refinement of each model.

We tuned the hyper-parameters of each model, taking into account their performance and avoiding over-fit, by preferring values with lower training set performance when confronted with values sharing the same testing set performance.

1) Linear regression: There were no hyper-parameters to tune.

2) Gradient boosting:

Loss function Combination of least squares regression and least absolute deviation;
Number of estimators 75 decision trees;
Max depth Depth of 2;
Number of samples to split 10 samples;
Number of samples to leaf 10 samples;
Max number of features All the 51 features.

3) Random forest:

Criterion Mean absolute error;
Number of estimators 25 decision trees;
Max depth Depth of 5;
Number of samples to split 2 samples;
Number of samples to leaf 40 samples;
Max number of features All the 51 features.

4) K-nearest neighbours algorithm:

Number of neighbours 12 units;
Leaf Size 15 units;
P 1,0;
Weights approach Uniform based;
Algorithm K-d tree.

5) Support-vector machine:

Kernel Radial basis function;
C 500;
ε 0,01.

VI. Results

Lastly, we compared the tuned models:

Their predictions were plotted against our testing set:

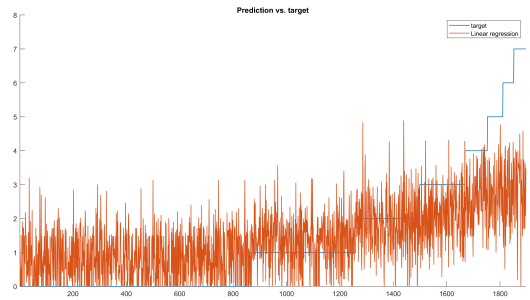

Fig. 6: Linear Regression Predictions vs. Target

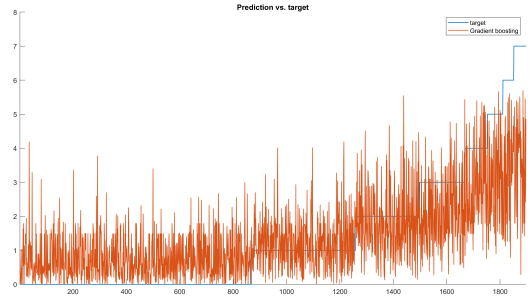

Fig. 7: Gradient Boosting Predictions vs. Target

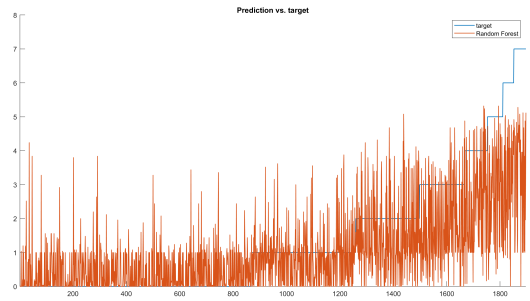

Fig. 8: Random Forest Predictions vs. Target

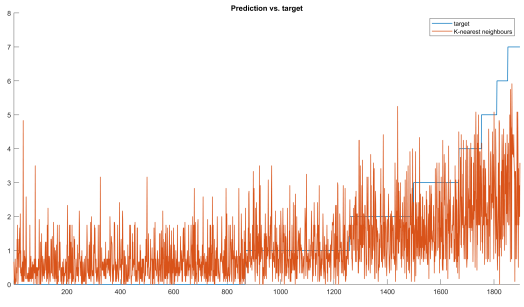

Fig. 9: K-nearest Neighbours Predictions vs. Target

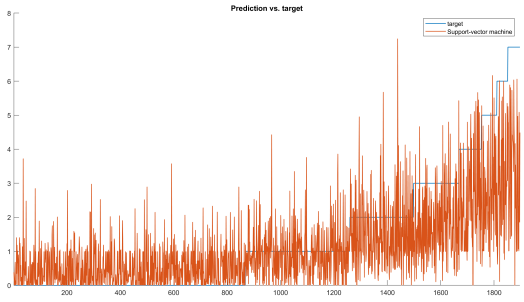

Fig. 10: Support Vector Machine Predictions vs. Target

VII. Conclusions

Taking into account the figures 4 and 5 and the differences between them, the random forest shows the best performance in our data set. Furthermore, every model yielded better results after refining their hyper-parameters, although with distinct amounts of improvement. Particularly, the random forest model presented a drastic change in performance, going from second-last to first.

As we can see from figure 6, linear regression takes a very conservative approach, overshooting low values and undershooting high values, which suggests the problem is not linear.

In other hand, gradient boosting [fig. 7] does a very good approximation of the lower values, consistently avoids negative values and accompanies the target line.

Similar to gradient boosting, random forest [fig. 8] goes further in avoiding negative values, with no negative values registered. However, the performance diminishes when approaching higher values, due to the conservative predictions.

From the figure 9, we can see that, although k-nearest neighbours is second-last in performance, it presents a relatively low variance.

Although the support vector machine [fig. 10] shows some similarity to our target, the results present a very high variance with extreme values for positive and also negative sides.

VIII. Future Work

In order to improve and continue this work, we think that a more complete data set, with more items from IPIP and from Malaise Inventory would provide more information and thus a more accurate prediction performance. An interesting strategy may be transforming our data into linear values, to perform linear approaches. A more autonomous search for the optimal hyper-parameters can also provide more interesting conclusions. Furthermore, more training models can be tested and tuned, such as linear regression variations, like Gradient Descent methods, which we implemented but didn't verify hyper-parameters. Finally, Locally Interpretable Model-agnostic Explanations (LIME) can be used to explain individual predictions for models[12].

In our opinion, gradient boosting shows the most desirable behaviour, with low variance, good curve, and some potential for further refinement.

IX. Acknowledgements

We want to thank the excellent documentation and framework of *scikit-learn*, which was the basis of this work and this paper.[13] Furthermore, we are thankful for the excellent insight and guide of Will Koehrsen with their machine learning project walkthrough.[14]

References

[1] Corr, Philip J.; Matthews, Gerald (2009). The Cambridge handbook of personality psychology (1. publ. ed.). Cambridge: Cambridge University Press. ISBN 978-0-521-86218-9.

[2] World Health Organization. Depression and Other Common Mental Disorders: Global Health Estimates WHO reference number: WHO/MSD/MER/2017.2

[3] Costa, Paul & Mccrea, R.. (1992). The Five-Factor Model of Personality and Its Relevance to Personality Disorders. Journal of Personality Disorders. 6. 10.1521/pedi.1992.6.4.343.

[4] C. Stachl, S. Hilbert, J.-Q. Au, D. Buschek, A. De Luca, B. Bischl, H. Huss- mann, and M. Buhner. Personality traits predict smartphone usage In *European Journal of Personality*, vol. 31, no. 6, pp. 701-722, 2017.

[5] Goldberg LR. A broad-bandwidth, public domain, personality inventory measuring the lower-level facets of several five-factor models. In: Mervielde I, Deary I, De Fruyt F, Ostendorf F, editors. Personality Psychology in Europe, Vol. 7 (pp. 7-28). Tilburg, The Netherlands: Tilburg University Press. 1999.

[6] Rodgers B, Pickles A, Power C, et al. Validity of the Malaise Inventory in general population samples. Soc.Psychiatry Psychiatr.Epidemiol. 1999;34:333-41.

[7] Hastie, Trevor; Tibshirani, Robert; Friedman, Jerome (2008). The Elements of Statistical Learning (2nd ed.). Springer. ISBN 0-387-95284-5.

[8] Allen TA, Carey BE, McBride C, Bagby RM, DeYoung CG, Quilty LC. Big Five aspects of personality interact to predict depression. Journal of Personality. 2018;86:714–725. https://doi.org/10.1111/jopy.12352

[9] University of London, Institute of Education, Centre for Longitudinal Studies. (2012). National Child Development Study: Sweep 8, 2008-2009. [data collection]. 3rd Edition. UK Data Service. SN: 6137, http://doi.org/10.5255/UKDA-SN-6137-2

[10] Hakulinen, C., Elovainio, M., Pulkki-Råback, L., Virtanen, M., Kivimäki, M. and Jokela, M. (2015), PERSONALITY AND DEPRESSIVE SYMPTOMS: INDIVIDUAL PARTICIPANT META-ANALYSIS OF 10 COHORT STUDIES. Depress Anxiety, 32: 461-470. doi:10.1002/da.22376

[11] Gareth, James; Witten, Daniela; Hastie, Trevor; Tibshirani, Robert (2015). An Introduction to Statistical Learning. New York: Springer. p. 315. ISBN 978-1-4614-7137-0.

[12] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16). Association for Computing Machinery, New York, NY, USA, 1135–1144. DOI:https://doi.org/10.1145/2939672.2939778

[13] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011. http://jmlr.csail.mit.edu/papers/v12/pedregosa11a.html

[14] Will Koehrsen's Machine Learning Project Walkthrough, acessible with the following link: https://github.com/WillKoehrsen/machine-learning-project-walkthrough