Cognitive, Behaviour and Social Data

Giovanni Dal Mas
MAT:2053346

Leonardo Schiavo
MAT: 2055519

Wu Xianlong
MAT: 2038500

Horatiu Andrei Palaghiu
MAT: 2050573

January 16, 2023

Abstract

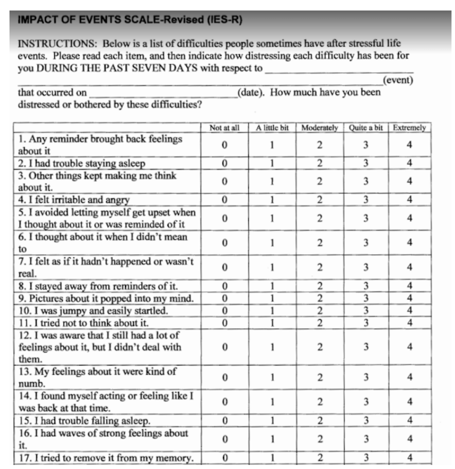
In this project, we are trying to gain insight into the impact different feature selection methods have on classification algorithms. Our core idea is to implement the most used variable extraction methods, both model dependent and model agnostic, on 13 different datasets. We then compare our results with a number of tuned classification models. Using psychological tests with varying number of items, we look at the impact the calculations done on these studies have on the questions retained on the final version of the tests. Hence, our main goal involves searching for techniques that have a good trade-off between replicability and result quality, by showcasing the relationship between model evaluation metrics and the correlation coefficient of highlighted features.

1 Introduction

In psychrometrics (the science of psychological test building), identifying a metric which emphasises the stability of the selected relevant questions in tests can be quite a challenge. Different methods are better suited for different questionaires, and the nature of dishonest answers is dependent on the intention of the participants. Thus, choosing a selection algorithm that is generally not so conditioned by the model might prove more reliable in consistently delivering similar results.

2 Datasets description

A number of studies have been conducted on volunteers, in which they have to answer honestly and dishonestly to a various questions regarding personal traits. These include, but are not limited to, the Dark Triad, five dimensions of human personality, signs of PTSD and memory impairment.

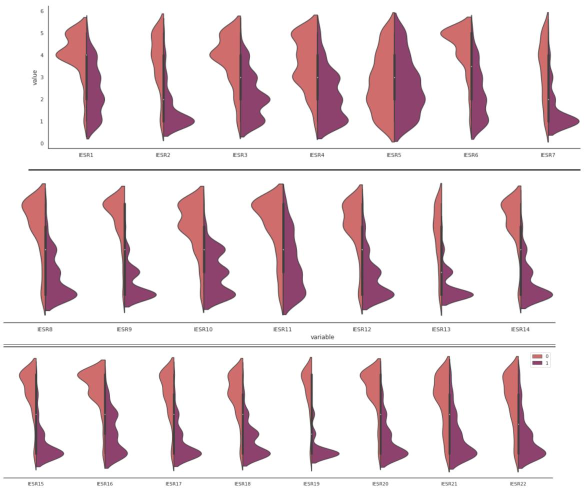


IMPACT OF EVENTS SCALE-Revised (IES-R)

INSTRUCTIONS: Below is a list of difficulties people sometimes have after stressful life events. Please read each item, and then indicate how distressing each difficulty has been for you DURING THE PAST SEVEN DAYS with respect to ____________________ (event) that occurred on ______________(date). How much have you been distressed or bothered by these difficulties?

	Not at all	A little bit	Moderately	Quite a bit	Extremely
1. Any reminder brought back feelings about it	0	1	2	3	4
2. I had trouble staying asleep	0	1	2	3	4
3. Other things kept making me think about it.	0	1	2	3	4
4. I felt irritable and angry	0	1	2	3	4
5. I avoided letting myself get upset when I thought about it or was reminded of it	0	1	2	3	4
6. I thought about it when I didn't mean to	0	1	2	3	4
7. I felt as if it hadn't happened or wasn't real.	0	1	2	3	4
8. I stayed away from reminders of it.	0	1	2	3	4
9. Pictures about it popped into my mind.	0	1	2	3	4
10. I was jumpy and easily startled.	0	1	2	3	4
11. I tried not to think about it.	0	1	2	3	4
12. I was aware that I still had a lot of feelings about it, but I didn't deal with them.	0	1	2	3	4
13. My feelings about it were kind of numb.	0	1	2	3	4
14. I found myself acting or feeling like I was back at that time.	0	1	2	3	4
15. I had trouble falling asleep.	0	1	2	3	4
16. I had waves of strong feelings about it.	0	1	2	3	4
17. I tried to remove it from my memory.	0	1	2	3	4

Then, regardless of the topic, researchers try to select the most effective questions in order to pass them to the final version. While some questionaries have very obvious skewness in distribution of the dishonest answers *(as the violin plots below of the labeled answer distribution on the PTST dataset show)*, some have a very hard to notice difference from performing EDA alone.



When we took a closer look at the datasets from a statistical point of view, we find that different reduction techniques provide sometimes drastically inconsistent results on whether some combination of questions might prove friendlier towards lie detection.

3 Classification procedures

We construct a lazy modelling function that integrates the most well-known statistical and machine learning tools for binary classification, which returns either an optimised model, coupled with a dictionary of already-tuned hyperparameters, or the expected and predicted values for the labels, ready to be passed forward for evaluation.

3.1 K Nearest Neighbours

Using only the notion of distances between datapoints, **KNN** distinguishes between groups of individual answers by a measure of similarity to already-labeled data. In spite of its simplicity, it proves highly effective on certain sets.

3.2 Support Vector Machine

Also known as just **SVM**, this algorithm focuses on constructing an optimal hyperplane that serves as a border between the half-spaces in which each labeled group might reside. In order not to overfit the training data, one must optimize on the flexibility of missclasification through the penalty term C.

3.3 Decision Trees

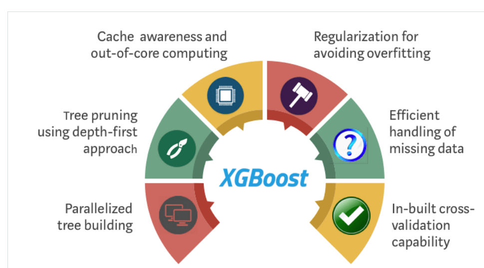
Structurally based on a sequential decision process, **decision trees** are quite intuitive, but somewhat predisposed to overfitting, making it necessary to use cross-validation and a maximum allowed depth. However, balanced 1:1 ratio between the labels of our data makes it favorable for avoiding bias.

3.4 Random Forrest

An ensemble-based learning method that extends on the principle of decison trees, **random forrests** are built under the concept of bootstrap-aggregation. While still staying under a desirable execution time for our small datasets, they prove to be very consistent in results, while not being affected by outliers or the non-linearity of the relationship between labels.

3.5 Neural Networks

While there exists an astonishing number of highly effective networks in the documentation, we went with the **Extreme Gradient Boosting** algorithm, also known as **XGB**, since the literature guarantees that it outperforms most popular architectures. Moreover, this is also part of what one could expect to see used for classification in many papers nowadays: the more reason its consistency in interaction with feature selection methods should be put to the test.

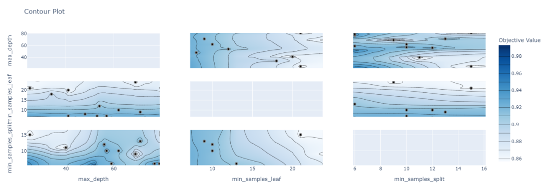


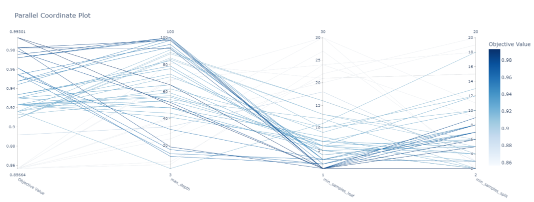
3.6 Implementation

Our modelling function works with all of the pre-processed datasets, and uses all the models mentioned above, optimizing them in order to maximize the score obtained through a k-fold cross-score validation procedure. Moreover, it can support both model-agnostic and model-depended feature selection algorithms.

4 Hyperparameter tuning

Due to the large number of unique datasets, model architectures, and selection methods that our implementation required, we decided to discard the classical Grid-Search algorithm for hyperparameter tuning. Instead, we opted for the more effective and versatile **Optuna** optimizaiton framework: it uses *TPESampler* as a base sampling algorithm, thus drastically reducing the time for finding the best hyperparameters.



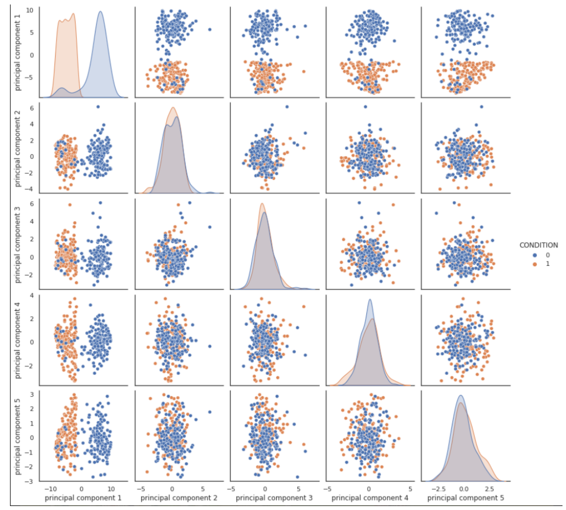


As shown in the images above *(of the contour and parallel search plots of our hyperparameter search for decision tree on the PTSD dataset)*, this library makes it very easy to identify and visually look for the areas with the best imput for our models.

5 Feature selection methods

5.1 Principal Component Analysis

PCA determines the optimal orthonormal linear combinations of features that describe the most percentage of data variance, also known as *factors*. However, it does not produce a list of columns that have the most impact on the factors.



5.2 Select K Best

The **Select K Best** method is a method that outputs the features according to the K highest scores produced by the score function. In this report, the **Chi-Square test** is used as the score function which tells us the **Chi-Square score** between the feature and the target.

5.3 LASSO applied to logistic regression

LASSO(least absolute shrinkage and selection operator) gives a L1 penalization to the problem and thus forcing a sparse solution. In our case, we will apply LASSO to the logistic regression as the problem is a binary classification.

5.4 LASSO applied to SVM

The base model we implement is **SVM(Support Vector Machine)** which maximizes the distance between two classes. And again with LASSO, we can obtain a sparse solution by adding a L1 penalty.

5.5 Variance Threshold Method

With the **variance threshold** method, we remove the features with low variance and keep the ones that explain the most variance in the data.

Thus, this method substitutes feature selection by PCA, since factors are created such that they represent most variance in the data *(this will be discussed in more detail later on).*

5.6 Correlation with target

The **Correlation with target** exploits the relationship between the independent and the response variables. The ones that have a higher correlation with the response is kept.
In order to implement this method, a pairwise correlations between each independent variables and the response are calculated, and then ordered in terms of the absolute value of the coefficients and in the end output the top 20 percent of the features as the top 20 percent most relevant ones.

5.7 Permutation Importance

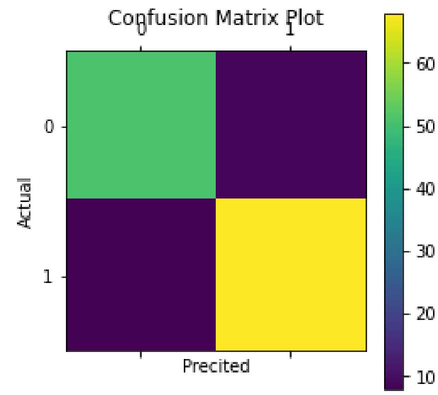
Permutation importance is a method for feature selection that involves randomly shuffling the values of a single feature and measuring the change in the model's performance. If the performance decreases significantly, it suggests that the feature is important to the model's predictions. This process is repeated for all features, and the features that result in the greatest decrease in performance are considered the most important.

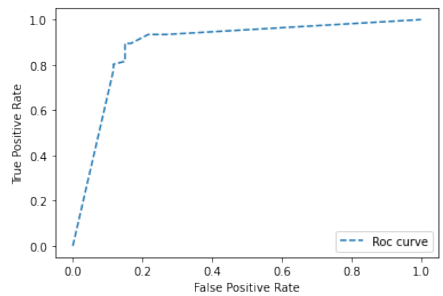
6 Experimets

6.0.1 Evaluation function

Within this function, various evaluation score have been implemented to access the performance of the experiment. The score used are: **Accuracy**, **Precision**, and **Recall**. The Accuracy takes simply the ratio between the number of correct prediction and the total number of predictions performed. The Precision takes the ratio between the number of true positives and the sum of true positives and false positives which is a measure of how many positive prediction made are correct. And finally, the last metric that is considered is the Recall or is also called Sensitivity, which measures how many ground truth labels has been predicted correctly.

There are also plots implemented in this function and the plots considered are: **Confusion Matrix** and the **ROC(Receiver Operating Characteristic)** curve together with the **AUC(Area Under the Curve)** score added.





The confusion matrix is a visualization tool such that its rows represents the the actual classes while its columns represent the prediction classes. And the ROC curve shows the trade-off between the Sensitivity and **Specificity** in which Sensitivity is already defined above and the Specificity is really similar to it but refers to the ground false labels. And AUC is just the area of the part that is under the ROC curve, and the higher AUC score is, the more accurate the model is.

6.1 Preliminaries

The data has been pre-processed and cleaned. Then we construct a function that has a dataset as input, which performs the train-test split, accompanied by normalization of all numeric entries and one-hot encoding the labels for faster calculations.

6.2 Results

Below one can see the accuracy averaged on a 5-fold cross-validated score on all 13 dataframes, for all models (highest accuracy highlighted).

No.	KNN	DT	RF	SVM	XGB
1	0.58	0.59	0.63	**0.64**	0.57
2	0.76	0.83	**0.87**	0.81	**0.87**
3	0.69	0.67	0.85	0.81	**0.87**
4	0.82	0.85	**0.96**	0.93	0.89
5	0.89	0.91	**0.97**	0.95	0.95
6	0.86	0.89	0.95	**0.96**	**0.96**
7	0.83	0.92	**0.95**	0.92	0.94
8	0.86	0.87	0.88	**0.90**	0.89
9	0.66	0.77	**0.94**	0.93	0.91
10	0.89	0.87	0.90	**0.91**	0.90
11	0.72	0.70	**0.79**	0.72	0.75
12	0.70	**0.77**	0.75	0.74	0.74
13	0.79	0.78	**0.86**	0.78	0.82

6.2.1 Accuracy shrinkage with respect to a fixed percent of data

In order to simulate how selection is done in reality in psychometric tests, each of our selection methods takes a fixed 20% of the features, to pass then again to evaluation. This restriction might come natural to the untrained eye (we want to select only X questions from a dummy test on to an official one), but statistically speaking, some models prove to be more sensitive (in terms of performance reduction) to the amount of shrinkage we perform on an individual questionnaire.

The selection method that by far proves to be more consistent in results (*meaning it has the most consistent accuracy drop across all datasets within* $+/-3\%$) is **SelectKBest**, as shown in some of the output our code provides (*pictures below*).

```
dataframe is: df_1
selection type: SelectKBest
Accuracy for KNNis 0.5185185185185185
----------------------------------------
Accuracy for Treeis 0.5962962962962963
----------------------------------------
Accuracy for RFis 0.5851851851851851
----------------------------------------
Accuracy for SVMis 0.5814814814814815
----------------------------------------
Accuracy for XGBis 0.5666666666666667
----------------------------------------

----------------------------------------
dataframe is: df_2
selection type: SelectKBest
Accuracy for KNNis 0.8434163701067615
----------------------------------------
Accuracy for Treeis 0.7758007117437722
----------------------------------------
Accuracy for RFis 0.7829181494661922
----------------------------------------
Accuracy for SVMis 0.7046263345195729
----------------------------------------
Accuracy for XGBis 0.7615658362989324
----------------------------------------

----------------------------------------
dataframe is: df_3
selection type: SelectKBest
Accuracy for KNNis 0.8148148148148148
----------------------------------------
Accuracy for Treeis 0.8148148148148148
----------------------------------------
Accuracy for RFis 0.8518518518518519
----------------------------------------
Accuracy for SVMis 0.8271604938271605
----------------------------------------
Accuracy for XGBis 0.8024691358024691
----------------------------------------
```

```
----------------------------------------
dataframe is: df_9
selection type: SelectKBest
Accuracy for KNNis 0.8194444444444444
----------------------------------------
Accuracy for Treeis 0.8472222222222222
----------------------------------------
Accuracy for RFis 0.8888888888888888
----------------------------------------
Accuracy for SVMis 0.8611111111111112
----------------------------------------
Accuracy for XGBis 0.8472222222222222
----------------------------------------

----------------------------------------
dataframe is: df_11
selection type: SelectKBest
Accuracy for KNNis 0.6464646464646465
----------------------------------------
Accuracy for Treeis 0.6515151515151515
----------------------------------------
Accuracy for RFis 0.6515151515151515
----------------------------------------
Accuracy for SVMis 0.6363636363636364
----------------------------------------
Accuracy for XGBis 0.6464646464646465
----------------------------------------

----------------------------------------
dataframe is: df_13
selection type: SelectKBest
Accuracy for KNNis 0.7194244604316546
----------------------------------------
Accuracy for Treeis 0.6870503597122302
----------------------------------------
Accuracy for RFis 0.6906474820143885
----------------------------------------
Accuracy for SVMis 0.658273381294964
----------------------------------------
Accuracy for XGBis 0.5071942446043165
```

6.2.2 The effect of methods on similar performance models

Most datasets have pairs of models that perform almost identically in terms of accuracy (less than 0.01 difference on a scale from 0 to 1). We than take these models and compare all the selection procedures mentioned above, showing the contrast between accuracy and the selected questions.

From the table on the top of the next page, we notice that for some models with equal accuracy, some methods give the same amount of shrinkage in accuracy (*all the cells in blue*), while the most offer inconsistent result. On the other hand there are many feature selection methods that paired with some models, offer an increase in accuracy (*the yellow cells*). Unfortunately, as much as we looked, there is no such thing as a consistent result. Every dataset has its own bias on which combinations they prefer. Moreover, what we noticed is that if one dataset tends to decrease consistently with one selection method, it will keep that consistency with an error $< 3\%$ with most other selectors.

No.	Model	Original	SelK Best	SVM Lasso	Log Lasso	Var Sel	Tar Sel
2	RF	0.87	0.77	0.77	0.77	0.86	0.74
	XGB		0.77	0.76	0.77	0.84	0.75
5	SVM	0.95	0.63	0.70	0.70	0.94	0.73
	XGB		0.73	0.74	0.75	0.82	0.83
6	SVM	0.96	0.92	0.97	0.96	0.96	0.89
	XGB		0.93	0.95	0.95	0.97	0.93
7	DT	0.92	0.89	0.71	0.87	0.93	0.77
	SVM		0.49	0.71	0.88	0.85	0.67
11	KNN	0.72	0.64	0.64	0.64	0.70	0.58
	SVM		0.64	0.64	0.64	0.72	0.56
13	DT	0.78	0.68	0.74	0.80	0.75	0.77
	SVM		0.62	0.71	0.79	0.79	0.67

6.2.3 Fake good versus fake-bad

Dishonest answers are an exaggeration of the truth, but they can take many forms. In our case, some datasets provide "faking good" answers(*those being 1,8,11 and 13*), and some "faking bad"(*those being 2,3,4,5,6,7,9 and 12*), depending on the imagination exercise participants had to do while lying in their responses.

For some reason, the first thing we noticed(from the first table and our confusion matrices), is that on average, our models detect faking bad answers much better that faking good.

6.2.4 Model-agnostic vs model-dependent selection

We found that surprisingly, model-dependent selection does not perform necessary better than model-agnostic, especially when the number of features to be kept is constrained. Take, for example, **L1 regularization**: model dependency makes it consider different sets of features, depending on the choice of model, while sometimes having a big drop in accuracy, precision and recall, due to the constraint rather on the number of factors, than on its hyperparameter.

6.2.5 Comparison with factor analysis methodology

PCA takes into account an orthonormal space, with directions formed by linear combinations of features that explain the most variance in the data. This is why it is very important to standardize it, so that a column with a bigger range of numbers does not weight too much on the final result. By performing PCA and then selecting/ranking the features that have the most absolute weight in the creation of the factors, one basically uses an approximation of **VarSelect**.

```
dataframe is: df_1
selection type: VAR_SEL
Accuracy for KNNis 0.5037037037037037
----------------------------------------
Accuracy for Treeis 0.5592592592592592
----------------------------------------
Accuracy for RFis 0.6074074074074074
----------------------------------------
Accuracy for SVMis 0.6481481481481481
----------------------------------------
Accuracy for XGBis 0.6333333333333333
----------------------------------------
----------------------------------------
dataframe is: df_2
selection type: VAR_SEL
Accuracy for KNNis 0.7366548042704626
----------------------------------------
Accuracy for Treeis 0.8113879003558719
----------------------------------------
Accuracy for RFis 0.8825622775800712
----------------------------------------
Accuracy for SVMis 0.8042704626334519
----------------------------------------
Accuracy for XGBis 0.8256227758007118
----------------------------------------
----------------------------------------
dataframe is: df_3
selection type: VAR_SEL
Accuracy for KNNis 0.6666666666666666
----------------------------------------
Accuracy for Treeis 0.7160493827160493
----------------------------------------
Accuracy for RFis 0.8395061728395061
----------------------------------------
Accuracy for SVMis 0.8888888888888888
----------------------------------------
Accuracy for XGBis 0.7654320987654321
----------------------------------------
----------------------------------------
```

```
----------------------------------------
dataframe is: df_11
selection type: VAR_SEL
Accuracy for KNNis 0.7171717171717171
----------------------------------------
Accuracy for Treeis 0.7272727272727273
----------------------------------------
Accuracy for RFis 0.797979797979798
----------------------------------------
Accuracy for SVMis 0.7121212121212122
----------------------------------------
Accuracy for XGBis 0.7474747474747475
----------------------------------------
----------------------------------------
dataframe is: df_12
selection type: VAR_SEL
Accuracy for KNNis 0.6777777777777778
----------------------------------------
Accuracy for Treeis 0.7444444444444445
----------------------------------------
Accuracy for RFis 0.7444444444444445
----------------------------------------
Accuracy for SVMis 0.6444444444444445
----------------------------------------
Accuracy for XGBis 0.7555555555555555
----------------------------------------
----------------------------------------
dataframe is: df_13
selection type: VAR_SEL
Accuracy for KNNis 0.8129496402877698
----------------------------------------
Accuracy for Treeis 0.8309352517985612
----------------------------------------
Accuracy for RFis 0.8525179856115108
----------------------------------------
Accuracy for SVMis 0.802158273381295
----------------------------------------
Accuracy for XGBis 0.8669064748201439
```

While the consistency of the list of selected features (*measured by Pearson's coefficient*) heavily depends on the specific dataframe, upon close examination we noticed that it preserves the most accuracy when put through a **Random Forrest** or **ExtremeGradientBoost** algorithm.

7 Conclusion and further research

All in all, this project proves it is more than understandable why there is an on-going debate on the state of experiment reproductibility, as different approaches to the same problem, and why making a choice on the trade-off between performance and result reliability make it hard to reach a consensus.

This is exactly why one should always consider using an ensemble of statistical tools and averaging the results when wanting to obtain a contradiction-proof result, especially in areas such as psychometrics: where the data is still growing, and the subjects could be prone to falsifying their evaluation.

References

Papers: 13 datasets

"Introducing the Short Dark Triad (SD3): A Brief Measure of Dark Personality Traits" by Daniel Nelson Jones & Delroy Paulhus - 2013.

"PMRQ - Questionario sulla memoria prospettica e retrospettica" by Smith, Della Sala, Logie, Maylor - 2000.

"The Posttraumatic Stress Disorder Checklist for DSM-5 (PCL-5): Development and Initial Psychometric Evaluation" by Christy A. Blevins, Frank W. Weathers, Margaret T. Davis, Tracy K. Witte, and Jessica L. Domino - 2015.

"The world turns at 33 and 45: Defining simple cutoff scores for the Negative Acts Questionnaire–Revised in a representative sample" by Guy Notelaers & Ståle Einarsen - 2012.

"The PHQ-9 Validity of a Brief Depression Severity Measure" by Kurt Kroenke, Robert L. Spitzer, Janet B. W. Williams - 2001.

"A Brief Measure for Assessing Generalized Anxiety Disorder, The GAD-7" by Robert L. Spitzer, Kurt Kroenke, Janet B. W. Williams, Bernd Löwe - 2006.

"The Personality Inventory for DSM-5 (PID-5) – Adult" by the American Psychiatric Association - 2013.

"The parental reflective functioning questionnaire: Development and preliminary validation" by Patrick Luyten, Linda C. Mayes, Liesbet Nijssens, Peter Fonagy - 2017.

"The Impact of Event Scale - Revised (IES-R)" by Donna McCabe - 2019.

"The Dirty Dozen: A Concise Measure of the Dark Triad" by Peter K. Jonason & Gregory D. Webster - 2010.

"An Italian version of the 10-item Big Five Inventory: An application to hedonic and utilitarian shopping values" by Gianluigi Guido, Alessandro M. Peluso, Mauro Capestro, Mariafrancesca Miglietta - 2014.

Bibliography

"The Elements of Statistical Learning" by Trevor Hastie, Robert Tibshirani, Jerome Friedman - Springer - Second Edition - 2020.

"Pattern Recognition And Machine Learning" by Christopher M. Bishop - Springer Nature - 2011.

"Deep Learning" by Ian Goodfellow, Yoshua Bengio, Aaron Courville - MIT Press - 2016.