**Universidade do Minho**
Escola de Engenharia
Departamento de Informática

Miguel Campos Calafate Carneiro Perdigão

# A Machine Learning Approach to The Big Five Personality Test

December 2019

**Universidade do Minho**
Escola de Engenharia
Departamento de Informática

Miguel Campos Calafate Carneiro Perdigão

# A Machine Learning Approach to The Big Five Personality Test

Master dissertation
Master Degree in Computer Science

Dissertation supervised by
**Cesar Analide de Freitas e Silva da Costa Rodrigues**
**Bruno Filipe Martins Fernandes**

December 2019

## DIREITOS DE AUTOR E CONDIÇÕES DE UTILIZAÇÃO DO TRABALHO POR TERCEIROS

# ACKNOWLEDGEMENTS

Depois de finalizado o trabalho aqui exposto, é importante parar para pensar, refletir e agradecer àqueles que foram de grande importância para a conclusão e sucesso, não apenas desta dissertação mas de todo o percurso durante o mestrado em que este trabalho se enquadra. Sem este grande grupo de pessoas que me apoiou de certa forma, tal não teria sido possível, agradecendo assim o apoio de todas elas.

Gostaria de começar por agradecer aos meus orientadores e professores Cesar Analide e Bruno Filipe, por todo o tempo dispensado e confiança em mim e no meu trabalho, pelos seus conselhos, orientação do trabalho mas também nas suas palavras que me levaram a não desistir e assim terminar este capítulo da minha vida.

Agradeço a todos os meus colegas de mestrado, pela amizade, companheirismo e acolhimento após a minha chegada, um novo membro no grupo.

Um enorme agradecimento à minha amiga Ivone que foi desde o inicío uma força para mim e uma fonte de inspiração pela sua qualidade e percurso profissional mas também pela sua forma natural de ser.

Por fim, à minha familia. Agradeço do fundo do meu coração à minha familia. À minha mãe, Teresa, ao meu irmão, Tiago, e à minha namorada, Daniela, tenho que deixar o meu sincero OBRIGADO! Pela força, atenciosidade, carinho, compreensão e felicidade que me proporcionaram nos momentos mais difíceis, não apenas durante este trabalho ou mestrado, mas durante a minha vida. Bons e maus momentos, todos eles tiveram a sua importância para hoje estar onde estou. Muito obrigado!

A todos que estiveram comigo, deixo o meu mais sincero obrigado!

# STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration.

I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

# ABSTRACT

One of the most accurate personality assessments available is the Goldberg's *'The Big Five Personality Test'*, which measures the five OCEAN dimensions: Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism. This assessment is performed by presenting a total of forty adjectives requesting the subject to rate each word using a scale of 1 to 9 indicating whether it accurately (9) describes herself or not (1). Nonetheless, scientific research has shown that this test may, accurately, suggest personality traits such as aggressive reactions, work performance, fitness on specific expertise areas and also mental illnesses. However, one big disadvantage of this test, it simply takes too much time to perform, which can result on undesirable measurements. Indeed, several developments have been done in order to reduce the required effort to perform this test, an example is *The Mini Marker Test* by Saucier. This study aims to propose a viable shorter alternative to this by applying machine learning techniques, i.e., although measurement precision may be reduced, is it possible to build a much shorter version losing as little precision as possible by just requiring the subject to select the adjectives that characterise him the most?

For this study, it was developed a platform to collect data, requesting both the subject to rate each adjective but also to select those he most identifies with. With this, the available data contains both ratings and the selections of the words that most characterise the subject. Three different machine learning architectures are developed and tested. Both regression and classification approaches are considered. The main input for these architectures are the words selected by each evaluated subject. Data collected by this work showed to be insufficient, requiring the use of data augmentation techniques. For this, different versions are proposed, one including the use of frequent itemset mining techniques. The proposed machine learning architectures shown a very high precision, with an RMSE of around 7%. The results show the proposed solutions to be able to perform a shorter version of this test with a minimum precision loss. It was also possible to define a list of common sets of selected words. Further research can be performed mainly on two different streamlines, i.e., strength the data collection process and develop an even shorter version of this test.

*Keywords:* big five, data augmentation, data science, machine learning

# RESUMO

Uma das avaliações de personalidade mais precisas foi criada por Goldberg, chamada *'The Big Five Personality Test'*, que mede um total de cinco dimensões denominadas de OCEAN: *Openness*, *Conscientiousness*, *Extraversion*, *Agreeableness* and *Neuroticism*. A avaliação em causa é realizada apresentando um total de quarenta adjetivos a um individuo solicitando lhe que classifique cada uma das palavras usando uma escala de 1 a 9, indicando se esta o descreve de forma exata (9) ou não (1). Assim sendo, estudos científicos sugerem que este teste poderá, de forma precisa, indicar outros traços da personalidade, tais como reações agressivas, desempenho no trabalho, aptidão para áreas de especialidade e doenças mentais. No entanto, uma grande desvantagem deste teste, é que este pode ser demasiado extenso e demorado, podendo gerar resultados indesejados. Na verdade, múltiplos desenvolvimentos foram feitos de modo a reduzir o esforço necessário para a realização do mesmo. Este estudo pretende assim propor uma alternativa mais curta e viável aplicando técnicas de *machine learning*, isto é, apesar da precisão dos resultados poder ser degradada, é possível construir uma versão muito mais curta com o mínimo possível de degradação da qualidade dos resultados apenas solicitando ao sujeito que este selecione os adjetivos que melhor o caracterizam?

Para este estudo, foi desenvolvida uma plataforma para recolha de dados, solicitando ao individuo tanto para classificar cada adjetivo, usando a escala, como também para selecionar aqueles com que este mais se identifica. Assim, os dados disponíveis contém tanto as escalas como a seleção das palavras que mais caracterizam cada um dos sujeitos. Três diferentes arquiteturas de *machine learning* são desenvolvidas e testadas. Tanto abordagens de regressão como classificação são consideradas. O principal *input* para estas arquiteturas é a seleção de cada uma das palavras por parte dos sujeitos avaliados. Os dados recolhidos durante este estudo demonstraram ser insuficientes, exigindo o uso de técnicas de *data augmentation*. Nesse sentido, diferentes versões são propostas, sendo que uma delas incluí o uso de técnicas de *frequent itemset mining*. As arquiteturas de *machine learning* propostas apresentaram uma precisão bastante elevada nos resultados, com um RMSE de cerca de 7%. Os resultados obtidos mostram que as soluções propostas são capazes de gerar uma versão reduzida do teste em causa com uma degradação mínima dos resultados. Foi também possível definir uma lista de conjuntos frequentes de palavras selecionadas. Desenvolvimentos futuros podem ser feitos em duas direções distintas, isto é, melhorar o processo de recolha de dados ou desenvolver uma versão ainda mais reduzida deste teste.

*Palavras chave:* big five, data augmentation, data science, machine learning

# CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LISTINGS

# ACRONYMS

**A**

**ANN** Artificial Neural Networks.

**APA** American Psychological Association.

**APPROACH I** Approach I - Multiple Independent Supervised Trait Regressors.

**APPROACH II** Approach II - Multiple Independent Supervised Scale Regressors.

**APPROACH III** Approach III - Supervised Multi-class Bin Classification.

**AUC** Area Under Curve.

**B**

**B5F** The Big Five Factors.

**C**

**CBR** Case Based Reasoning.

**CRISP-DM** Cross-industry standard process for data mining.

**CV** k-fold Cross-Validation.

**D**

**DA** Data Augmentation.

**F**

**F1** F1 Score.

**FIM** Frequent Itemset Mining.

**G**

**GB5P TEST** Goldberg's Big Five Personality Test.

**H**

**HR** Human Resources.

**HYPERPARAMETER TUNNING** Hyperparameter Optimization.

**M**

**MAE** Mean Absolute Error.

**ML** Machine Learning.

**MM TEST** The Mini-Marker Test.

**O**

**OCEAN** Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism.

**P**

**PCA** Principal Component Analysis.

**PHY. AGG.** Physical Aggression.

**PHY. AX.** Psychological Assessment.

**PP TEST** Phased Partial Test.

**R**

**RMSE** Root Mean Square Error.

**S**

**SVM** Support Vector Machines.

**T**

**TDSP** Team Data Science Process by Microsoft.

**W**

**WSP TEST** Word Selection Personality Test.

**X**

**XGBOOST** eXtreme Gradient Boosting.

# 1

# INTRODUCTION

In this chapter the motivation for this work is described along with a brief contextualisation of the main domain which this is applied. A full comprehensive list of all tackled objectives is listed with a brief description of each one. In the end, the structure of the entire document is presented with a description of each chapter.

## 1.1 CONTEXTUALISATION & MOTIVATION

The Human being has been studying himself for a long time, however, there is still plenty to discover and understand about us. One of the big mysteries scientists continue to study is our minds. Different people react differently to different situations, maybe because of each one's knowledge, experiences or other factors. However, one factor that surely induces people to take different approaches to a problem is personality. Some state of the art methods propose a connection between a person's personality and aggressive reactions. Different studies show a relationship between personality and work performance or fitness on specific working areas. Having this in mind, multiple studies are being performed to understand our minds.

If studies propose relationships between personality and reactions or characteristics, it means that it is possible to classify a person's personality, using a qualitative or a quantitative metric. Nowadays, there are several accepted tests which allow a psychological assessment of any person. These are currently being performed on different occasions such as psychology appointments, job interviews, psychometric tests (being these used by multiple entities, like companies, the army, the government, etc). However, these tests are mainly conducted by psychology professionals with the correct training to perform them and interpret their results. Another problem with these tests is the required time to perform this full assessment.

Machine Learning has been a really amazing tool, having so many attention by academia but also by media and the industry. However, Machine Learning should be seen as a mean to an end and not an end by itself. Having this in mind, several researchers and companies have been using Machine Learning to solve both ancient and modern problems, bringing new discovers/solutions and creating new products which streamline our daily lives. Inspired by those people, an approach to consider to these psychological tests involves using Machine Learning techniques, not in order to create new tests from scratch but to improve existing ones. Improving time consumption and the presence of specialised professionals, improving the ratio cost result.

## 1.2 OBJECTIVES

The main goal of this work is to develop a machine learning framework to identify a person's personality based on the choice of a subset of words from a total of 40 words with the same or as near as possible the reliability of the *The Mini-Marker Test (MM test)* [2]. The resulting framework is tested against different processes already accepted by the scientific community, being expected to verify if those processes are still valid using the developed framework. If so, this will allow to minimise the costs of Psychological Assessment tests such as The Mini-Marker Test.

In detail, this work's objectives are:

1. Collect all required data
   - Before being able to perform any type of study, analysis or applying Machine Learning to the Big Five Personality Test, it is required to collect as much data as possible. Different collecting processes are used and described in the next chapters.
2. Apply Pattern Mining on selected words
   - Analysis and implementation of pattern mining techniques to understand and document existing relations and patterns in the selection of words using the collected data.
3. Estimate a persons personality based on selected adjectives instead of ratings
   - After collecting and processing relevant data that describe answers to the Mini-Marker Test, the goal is to consider a Machine Learning approach to this problem developing a framework to compute models to estimate a persons personality based on selected adjectives instead of ratings.
4. Ensure the highest reliability as possible
   - From Goldberg to Mini-Marker factors set, reliability is one of the reasons this new set does not totally replaces the original set, according to original paper. Therefore, it is important to keep reliability as higher as possible.

5. Validation of the developed framework against state of the art methods
    - It important to evaluate the relationship between this framework results and the original Goldberg's Big Five Personality Test performed in order to understand if this can partially or even fully replace.
6. Propose a Phased Partial Test
    - Try to use not 40 adj. but just the absolute necessary. For example, start by showing a word and according to its selection or not, different words(s) will appear to the user. For example, using this methods we could be able to reduce from 40 adj. to only 5 (just an example for the explanation). It allow us to have an enormous complexity reduction. Although in this work this will not be implement, a suggestion and reference for future work is described.

## 1.3 STRUCTURE OF THE DOCUMENT

This document is organised in the following way:

**Chapter 2 - state of the art**

Introduction to some of the Physiological concepts and the available and most accepted/trusted assessments as well some of the studies performed on the relationship between these tests' results and Mental Disorders. It also contains a short introduction to some machine learning concepts and models and some useful statistical tests for data exploration.

**Chapter 3 - the problem and its challenges**

In depth description of the problem studied in this work, what are the different challenges of it and the proposed approaches to solve the described problems. It is also described what are the restrictions and limits of these approaches.

**Chapter 4 - model design and development**

Description and analysis of all made decisions during the development of this investigation and also a full description of the implementation of all proposed approaches in the previous chapter.

**Chapter 5 - experiments and results**

Full description, analysis, discussion and used setup of the perform investigation are covered. Although a in depth analysis is described a summary is also presented.

**Chapter 6 - conclusions and future work**

In this chapter the conclusions of the conducted study of the problem are presented. It also presents the next steps for future work.

# 2

# STATE OF THE ART

In the following chapter a full description of the relevant state of the art topics is presented. Firstly, the concept of Psychological Assessment is presented, a description of what it is, how it is performed and the requirements of these are explained. One of the main concepts of this work is then introduced, The Big Five Factors which is related to the previous topic. To understand the relevance and an important application of these concepts, the relation between B5F and Mental Disorders is briefly explained. In the end, the main methods used to solve the described problem are enumerated and discussed, including Machine Learning concepts and algorithms.

## 2.1 Psychological Assessment

*Psychological Assessment (Phy. Ax.)* is a well defined process which aims to infer hypothesis about a person, their behaviour, personality characteristics and day to day competences. This well defined process, is composed by a battery of tests and techniques. Usually, these tests are performed by trained psychologists which are prepared to correctly interpret the psychological results. According to the *American Psychological Association (APA)*, a Phy. Ax. can include numerous components, surveys, informal tests, interview information, medical or school records and norm-referenced psychological tests [1]. These kind of tests are important for many different reasons, not just for psychologists, but also for *Human Resources (HR)* professionals and job placements specialists. The results from these tests can reveal a lot about a persons personality, since it reflects on how people react to different situations. This can tell us what a person is meant to, what activities, tasks or jobs are appropriate and we can even notice if it is genetically inclined to have some sort of mental illness. For example, these tests can determine if a person would be good as a Doctor, Lawyer or a Manager. They also allows to determine cases of learning disorder, violent behaviours, anxiety or depression.

[1] *"Assessment that compares test-taker performances to each other, i.e. when students scores are ranked from low to high, and their rankings are compared to each others. There is no attempt to interpret the scores in terms of what students know and can do, except in the limited sense that a students performance is typical of other low, middle, or high performing students in the group."*, source definition: American Education Research Association (AERA), American Psychological Association (APA), and National Council for Measurement in Education (NCME) (1999)

## 2.2 Introduction to The Big Five Factors

Nowadays, there are multiple psychological tests to define a persons personality like HEXACO-60 [5], Myers-Briggs Type Indicator [6], Enneagram of Personality [7] and many others. However, *Goldberg's Big Five Personality Test (GB5P test)* [1], with a strong scientific origin, is one of the most accurate tests available [2].

*Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism (OCEAN)* are the big five personality traits (aka B5F) defined from the statistical study of responses to personality tests. The original GB5P test consists of 100 pre-selected adjectives that the respondent must rate on how true they are about himself on a 9 point scale, where 1 = extremely inaccurate and 9 = extremely accurate. As result of this test, we can measure the level of each one of the OCEAN personality dimensions. However, one of the biggest problems of GB5P test is the time consumption [2] and costs associated to perform this full psychological evaluation. Having this in mind, many studies were conducted in order to reduce the list of 100 adjectives as stated by Saucier [2]. Although the enormous amount of experiments developed to accomplish this goal, one of the main reductions was performed by Gerard Saucier with The Mini-Marker Test, determining a sub list of 40 adjectives, assessing OCEAN dimensions with almost the same performance.

## 2.3 Relation between B5F and mental disorders

Currently there are a variety of health problems, diseases and disorders. We can detect different type of diseases, a person may have one or more types of diseases. There are infectious diseases, deficiency diseases, hereditary diseases (including both genetic diseases and non-genetic hereditary diseases) and physiological diseases. Diagnose those diseases is really important, in some cases, it may be the difference between almost 100% changes of success to cure it or near 0% to accomplish it.

Some diseases may or may not have risks to a person life. However, they may get a person to perform certain actions which can lead to hurt a third person, in this particular case, physiological diseases or mental disorders are being referenced. For example, aggressive emotions and aggressive attitudes can lead to violent actions on a third person. Barlett and Anderson on a study of 2012 [8], suggests a relationship between Big 5 personality traits and aggressive behaviour directly and/or indirectly through emotions and attitudes. In this study, different relations were inferred, for example: 1) for *Physical Aggression (Phy. Agg.)* Aggreableness was indirectly negatively related to aggressive behaviour (both on emotions and attitudes); 2) also for Phy. Agg., Neuroticism was indirectly related to aggressive behaviour through aggressive emotions. Another study, by Kotov, R. on [11], a relationship between B5F and Personality Traits like Anxiety, Depression and Substance Use Disorderds can be established.

## 2.4 MACHINE LEARNING CONCEPTS AND ALGORITHMS

Machine Learning it is a powerful technique, however there are some important concepts to understand before applying them. In this section, a set of helpful steps to guide our work in the process of building Machine Learning models are presented. The difference between *Supervised and Unsupervised Learning* is described followed by an important description of the different types of learning methods. The tree based algorithm eXtreme Gradient Boosting it is also described. In the end, a comprehensive summary of this chapter is present.

### 2.4.1 *Steps to build a Model*

A methodology or process must be followed to ensure the success of a project, with examples being the *Cross-industry standard process for data mining (CRISP-DM)* [9] or the *Team Data Science Process by Microsoft (TDSP)* [10]. Having this in mind, there is a set of steps to perform to build the *Machine Learning (ML)* model that will be the foundations of the developed framework. The following steps on **Figure 1** are considered.

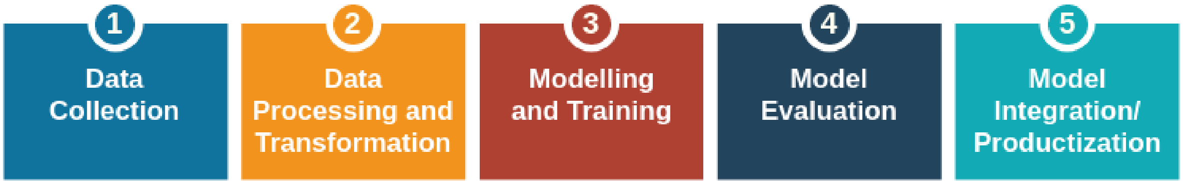

1
Data Collection
2
Data Processing and Transformation
3
Modelling and Training
4
Model Evaluation
5
Model Integration/ Productization

Figure 1.: 5 steps process to develop the a Machine Learning Model.

Any ML model needs of data in order to be trained. **step 1** intends to solve that need, where all needed data is gathered, documented, stored and proper access is guaranteed. **Step 2** dives deeper into data exploration and data transformation, using statistical tests, data wrangling techniques, feature engineering and other where it is pretended to take as much value as possible from data. In the **third step**, the model is trained, using the transformed data, and tuned, in order to end up with a better representation of the scenario described by the data. The performance of the trained model must be then evaluated, **step 4**, where different metrics and tests must be used according to each problem. These steps are iteratively performed to develop the best model possible. In the end (**step 5**), the picked model must be integrated to have the final usage, which could be a simple single usage, a product integration or a set of access points to it.

### 2.4.2 *Supervised and Unsupervised Learning*

Machine Learning consists on three main types of learning, supervised, unsupervised and reinforcement learning. Each one of these applies to different situations, with different characteristics. However, each one of them has specific characteristics and limitations. To notice that the third one, reinforcement learning, it is not relevant for the development of this work. While supervised learning uses a ground truth, where we have prior knowledge of the output values for the used samples, on unsupervised learning, we have no prior labelled outputs. The main goal of supervised learning is, by using sample of input and desired outputs, to learn a representative function of the relationship between input and output from the data. On the other hand, unsupervised learning, having no labelled outputs, its goal is to infer the natural organisation and structure of the presented data points.

Typically, supervised learning is performed in the context of classification and regression problems, map input to output labels and map input to continuous output respectively. Some common algorithms are Logistic Regression, Naive Bayes, *Support Vector Machines (SVM)*, *Artificial Neural Networks (ANN)* and Random Forests. Independently of classification or regression, the goal is to find specific relationships or structures in the input data allowing to produce correct corresponding output data. How correct is the output, is partially related to training data, which brings us to noisy or incorrect data. If we have a poor ground truth that our model assumes as true, it will clearly reduce the effectiveness of our model.

Some of the main considerations during this type of learning, is both model complexity and the bias-variance tradeoff. Complexity can be understood as the complexity of the function/relationship it is being attempted to learn, this can be determined by the nature of training data. In case of small datasets with low variance, low complexity models are preferred in order to avoid overfitting. On the other hand, big datasets with high variance, highly complex models are needed in order to represent all the difference. Bias-variance tradeoff is related to model generalization, the balance between constant error term (bias) and the amount the error can vary between different training sets (variance) [2]. Having this in mind, a balance must be assumed according to the needs. Generally, increasing bias allows to guarantee a firm baseline of performance levels which may be critic for different contexts.

On unsupervised learning, clustering or representation learning are common tasks. In this type of learning, it is pretended to learn the hidden or unknown structure of a specific dataset without explicitly providing labels. Some known common algorithms are k-means clustering, *Principal Component Analysis (PCA)*, DBSCAN or even autoencoders. Exploratory

[2] E.g.: High bias and low variance could be a model that is consistaly wrong 20% of the time; Low bias and high variance could be a model that can be wrong 10%-%30 dependeing on the data used to train

Analysis and Dimensionality Reduction are only two of common use-cases. For Exploratory Analysis, unsupervised learning allows to automatically identify structure in data, like users segmentation, where (most of the times) is humanely impossible to see trends in the data, allowing the test different hypothesis. On the other hand, Dimensionality Reduction [3] can also be conducted using unsupervised learning algorithms, allowing us to find a subgroup of characteristics of our data.

### 2.4.3 *Types of Learning Methods*

Machine Learning involves multiple concepts and techniques, being important to identify what type of problem we have in hands and what the most proper techniques for that problem. We can identify three main types of ML problems: 1) **Classification** Section 2.4.3; 2) **Regression** Section 2.4.3; 3) **Clustering** Section 2.4.3. All different types have their characteristics which makes them appropriated for specific problems.

#### *Classification*

Classification is a type of learning method or algorithm. This process predicts the class (label, target) of specific input data points, by formulating a mapping function between input variables to the discrete output variables.

Depending on the context, different outputs are expected from the models. The classification output can be of different types, **Binary** or **Multiclass classification**. Binary classification is characterised by a binary output, two classes to predict, usually 1 or 0. On the other hand, multiclass classification is characterised by having more than two class labels to predict.

This classification algorithms can be divided into main types, **Lazy** and **Eager Learners**. The first ones, stores the training data, when testing data appears, the classification process is conducted based on the stored data that most related to the new data. These algorithms have the advantage to be less time consuming on training but take more time during prediction. On the other hand, eager learners, build a classification model having training data as a solid base. Only then, this model receives new data to perform classifications. Although these take longer to train they are faster to predict. Some well known eager classification algorithms are Decision Trees, Logistic Regression, Naive Bayes and ANN. Examples of lazy learners are K Nearest Neighbours or *Case Based Reasoning (CBR)*.

[3] Techniques to represent data using less columns or features.

*Regression*

As described on **Section 2.4.3**, classification is the task of mapping a function between input variables to a discrete output. On the other hand, regression maps a function of input variables into a continuous output result, describing a relationship between a set of independent variables and a dependent variable. This continuous output can be any real-value, depending on the context.

Like classification, regression can also be used both for binary and multiclass categorical problems, however it presents a different process. In a multiclass problem, the 'one-vs-all' method is applied, which is simply to apply binary regression to each of the available classes, testing each class against all the others.

Although we have different algorithms, we must understand which ones are the best for the analysis we intend to perform, this is determined by the output dependent variable. We can have a continuous dependent variable, which can be linear or non-linearly described. To conduct this analysis, we can use linear methods like OLS, Linear Least Squares or to address some problems [4] of these like Ridge, Lasso or PLS regressions. Regression can also be used with categorical dependent variables as described before. In cases where OLS can not be used, because count distribution is some how skewed [5], we have an analysis of count dependent variables, having several models that can be used, like Poisson regression or Negative Binomial or Zero-inflated regression models.

Nowadays there are several regression algorithms, like Linear/Polynomial Regression, for continuous targets, Logistic Regression, for discrete output, SVM can also be applied, Decision Trees and Random Forest are also very common algorithms.

*Clustering*

Clustering is one of the possible types of unsupervised ML. The main essence of clustering is the capability to identify similar groups of data points of a dataset in respect or its attributes or characteristics. Each entity of each cluster (group) is comparatively more similar to entities of the same cluster than to the other ones. This process is performed by dividing the different data points into clusters (groups) by using a specific method to calculate the similarity of these entities. Points with similar traits are assigned to the same clusters. This can be useful in different problems and contexts, to get deeper insights of big data, to perform analysis on apparently unrelated data.

Clustering algorithms can be divided into two main groups, **Hard** and **Soft clustering**. While Hard clustering requires a data point to completely belong to a cluster or not belong

[4] This methods have several weaknesses, like sensitivity to outliers and mulicollinearity and it is prone to overfitting.

[5] Skewed data is not equally distributed on both sides of the distribution, so it is not normally distributed.

to it at all, Soft clustering assigns a probability to belong to a certain cluster, being this a huge advantage for some situations and analysis.

The method to calculate the similarities of the entities, varies according to the different algorithms and methodologies. Having this in mind, we can recognise four main methods. **Connectivity methods**, based on the notion of spatial distance, where the closer the similar they are. These can follow two approaches. The first is to start by dividing into multiple clusters and then aggregating them on distance decrease. The second approach is to consider all data points a single cluster and then partition it by distance increase. **Centroid methods** consider a iterative process, where the notion of similarity is defined by the spatial distance to the centroid [6] of the cluster. This methods require to have prior knowledge of the dataset, since its needed to indicate the number of clusters beforehand. **Distribution methods**, as the name indicates, these are based on the probability of all data points in the cluster belongs to the same distribution. **Density Models**, assigns clusters to areas of varied density of data points in the data space, isolating various different regions.

There are several clustering algorithms known, some of them are K Means Clustering, Hierarchical Clustering, Mean-Shift Clustering, DBSCAN and EM using GMM.

### 2.4.4 *eXtreme Gradient Boosting*

*eXtreme Gradient Boosting (XGBoost)* is a particular implementation of boosted trees, which presents several advantages and capabilities. A common technique used to avoid overfitting is called *regularization*, which prevents to learn a more complex model. Standard boosted tree based algorithms does not implement this technique natively. However, XGBoost uses this to reduce overfitting.

Parallel processing is one of the hottest topics, which allows faster processing. This is one of the advantages, this allows to use all available cores to build each of the boosted trees. However, since this algorithm builds consecutive trees based on the previous one, it is still only possible to build one tree at a time. Although not necessarily useful for this project, it is that XGBoost also supports implementation on Hadoop.This is highly flexible, allowing users to define custom optimisation objectives and evaluation criteria. Although it has to exist input from the user, XGBoost has natively supports missing values with a set of in-built routines. Per last, another advantage, is that any of these final trained models can still be re-trained at any moment, starting the training from its last iteration of previous run.

[6] The middle of a cluster. A centroid is a vector that contains one number for each variable, where each number is the mean of a variable for the observations in that cluster.

## 2.5 SUMMARY

Several studies have been conducted in the context of this work, both on *Psychological Assessment (Phy. Ax.)* and on *Machine Learning (ML)* techniques, what can allows us to affirm two things. In the first context, it is noticed there is a huge interest to improve and conduct even more studies in the area of Phy. Ax., in order to improve them both on accuracy and performance, allowing us to understand the human personality in a better and correct way. However, it is also important to have tests with better performance, by creating simple and less time consuming tests. On the context of ML, it is interesting to see the current developments and available tools we have to solve different problems. Having these developed algorithms and tools, it allow us to leverage current solutions to the next level, by using Machine Learning techniques we can improve current solutions and even create new ones. Therefore, it is important to denote the relevance of this work, having both the need to solve a problem and the techniques to do it, ”A Machine Learning Approach to the Big Five Personality Test” has a promising success of being completed and having a good impact on the way we perform the Big Five Personality analysis.

# 3 THE PROBLEM AND ITS CHALLENGES

## 3.1 PROBLEM DESCRIPTION

Psychological assessment is a very common test performed on several contexts. However, a great amount of time is associated with tests such as the *Goldberg's Big Five Personality Test (GB5P test)* and even *The Mini-Marker Test (MM test)*. MM test requires the user not only to analyse and process a total of 40 adjectives but also to classify each one of them in scale of 1 to 9. This can bring multiple consequences, it requires a large load of time and patience, increasing the possibility of fatigue or short attention span, resulting on undesirable measurements [2].

In **Figure 2**, Goldberg's Big Five Personality Test, consisting of a total of 100 adjectives, requiring the subjects to scale each one of them, allows to measure each one of the *Openness, Conscientiousness, Extraversion, Agreeableness and Neuroticism (OCEAN)*'s factors. Saucier on [2], proposed the MM test, which could reduce the set of 100 words into only 40 of them.

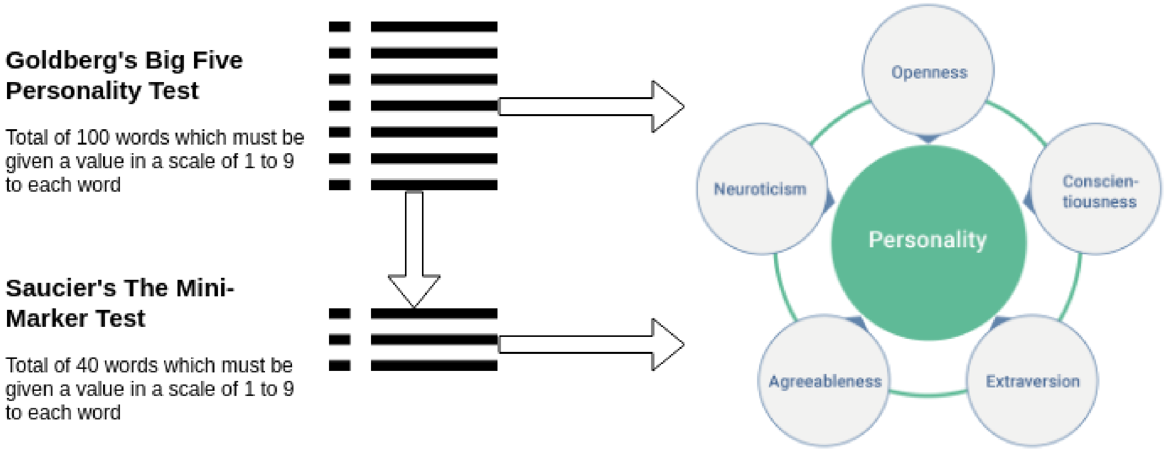

Goldberg's Big Five Personality Test
Total of 100 words which must be given a value in a scale of 1 to 9 to each word
Saucier's The Mini-Marker Test
Total of 40 words which must be given a value in a scale of 1 to 9 to each word
Openness
Neuroticism
Conscien-tiousness
Personality
Agreeableness
Extraversion

Figure 2.: Current accepted Goldberg's and Saucier's Personality Tests.

This work proposes two new approaches to reduce the previous tests. Although Saucier's reduction was on the level of the number of words, these approaches are applied to reduce the process of interaction with the subject. The proposed ***Word Selection Personality Test (WSP Test)*** in this work, **Figure** 3, suppresses the entire process of using a scale for each word into the simple selection of the words the subject identifies with. This change is intended to be accomplished by using different *Machine Learning (ML)* techniques described in **Section** 2.4 and **Section** 3.3.

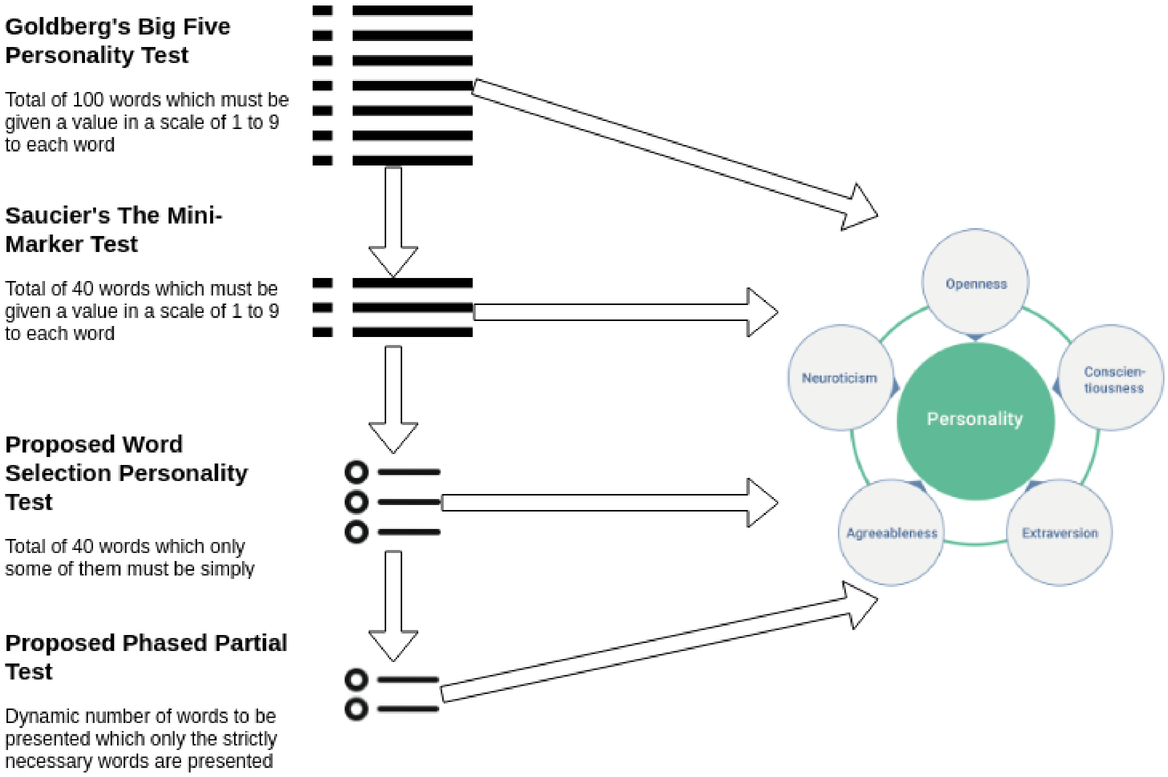

Goldberg's Big Five Personality Test
Total of 100 words which must be given a value in a scale of 1 to 9 to each word
Saucier's The Mini-Marker Test
Total of 40 words which must be given a value in a scale of 1 to 9 to each word
Proposed Word Selection Personality Test
Total of 40 words which only some of them must be simply
Proposed Phased Partial Test
Dynamic number of words to be presented which only the strictly necessary words are presented
Openness
Conscien-tiousness
Extraversion
Agreeableness
Neuroticism
Personality

Figure 3.: Proposed Personality Test: a) **Word Selection Personality Test** and b) **Phased Partial Test**.

An even deeper reduction of the WSP Test is also proposed in a second stage of this work, the ***Phased Partial Test (PP Test)*** in **Figure** 3. **Figure** 4 is a representation on a high point of view of PP Test's process. In this diagram, ***W(x)***, represents the words presented to the subject, where **'x'** is a representation of one or multiple words[1]. In this test, words shown to the subject are dynamic, which means, different subjects are subjected to different words, depending on their selection or not of the shown words. For example, as we can see on **Figure** 4, *w(1)* is shown, which the subject can select it or not: 1) *'Red arrow':* Subject does not selects any word; 2) *'Yellow arrow':* Subject selects *n* words out of *w* words, where

[1] **W(1)**: 'word one' is shown to subject; **W(5, 6, 7)**: 'word five', 'word six', 'word seven' are shown to subject.

o>n>w; 3) *'Green arrow':* Subject selects all words. According to the choice, *w(2)* or *w(3, 4)* is then shown. This is repeated until the proposed framework is able to calculate the OCEAN factor result. On an hypothetical optimal scenario, the subject is subjected to only 3 words, the *w(1)* and *w(3, 4)*, instead of the normal MM test's 40 words or even GB5P test's 100 words.

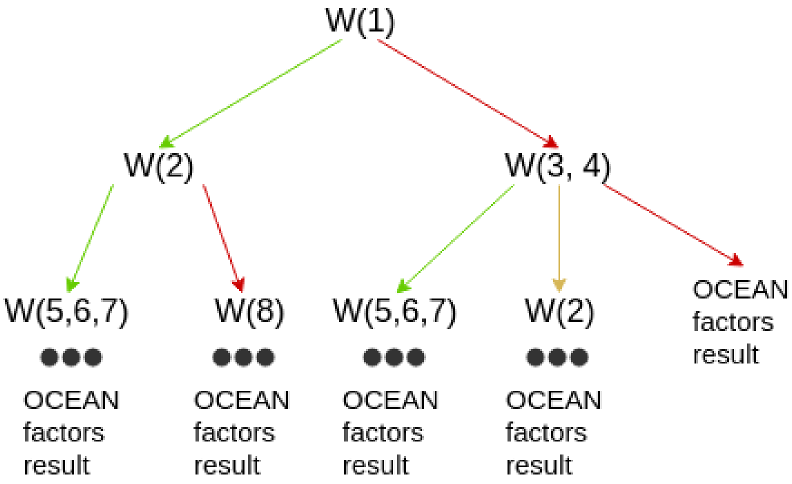


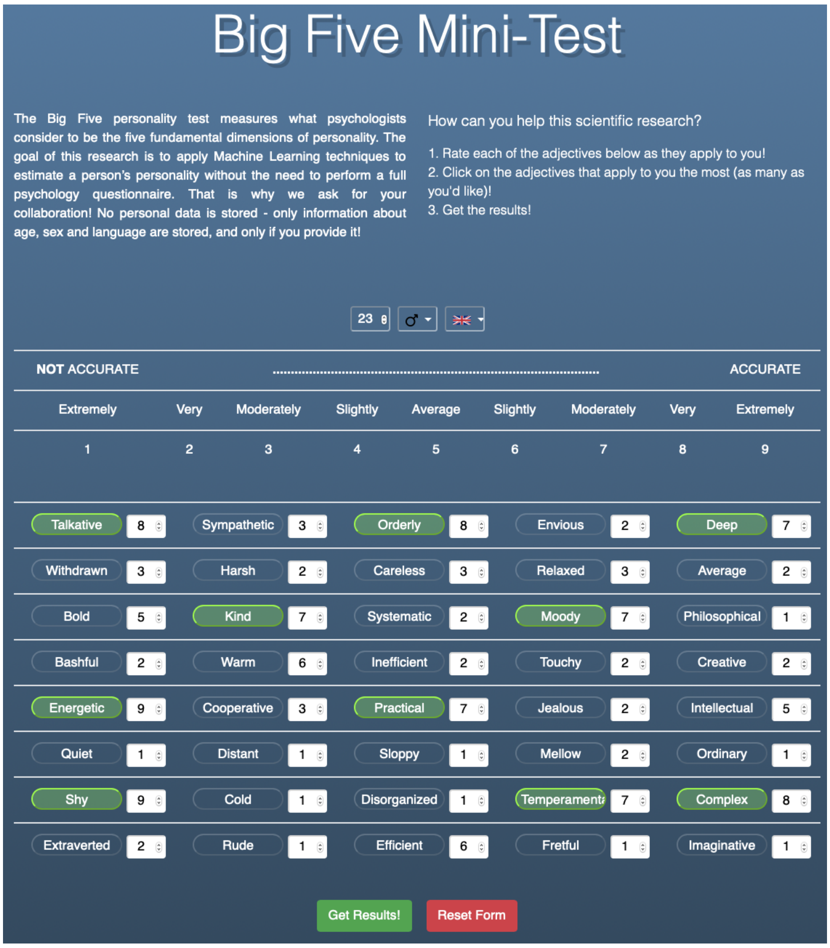
Figure 4.: Proposed **Phased Partial Test** design.

## 3.2 CHALLENGES

One of the main challenges to overcome, is that data is not available. As mentioned on **Section 2.4**, to train a ML model, we need great amounts of data. Although it may not seem that big of a problem on the XXI century, it requires a lot of time to collect a reasonable amount of data. Not just the time to collect this data, but also because it requires uncommon data and a population of subjects opened to give this data in a free will but also to ensure the data is correct, what brings us to the next point. The results of a model are related to the data the model uses to train, which requires the data to be filtered when data may not be correct, which can lead to a great reduction of the collected data.

Every single model created in this work is trained using data collected by a platform that was developed for such end. This platform, displays all of the 40 words, where the user/subject must use the MM test's scale for each of the words but also select the ones he identifies with. This platform is available online where any person can access and fill the form. This platform have been shared in order to collect data from a varied population. **Figure 5** is the main page of the developed platform available online[2].

[2] Platform available on http://crowdsensing.di.uminho.pt/



Big Five Mini-Test

The Big Five personality test measures what psychologists consider to be the five fundamental dimensions of personality. The goal of this research is to apply Machine Learning techniques to estimate a person's personality without the need to perform a full psychology questionnaire. That is why we ask for your collaboration! No personal data is stored - only information about age, sex and language are stored, and only if you provide it!

How can you help this scientific research?

1. Rate each of the adjectives below as they apply to you!
2. Click on the adjectives that apply to you the most (as many as you'd like)!
3. Get the results!

23 ♂

| NOT ACCURATE | | | | ........................................ | | | | ACCURATE |
|---|---|---|---|---|---|---|---|---|
| Extremely | Very | Moderately | Slightly | Average | Slightly | Moderately | Very | Extremely |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Talkative | 8 | Sympathetic | 3 | Orderly | 8 | Envious | 2 | Deep | 7 |
| Withdrawn | 3 | Harsh | 2 | Careless | 3 | Relaxed | 3 | Average | 2 |
| Bold | 5 | Kind | 7 | Systematic | 2 | Moody | 7 | Philosophical | 1 |
| Bashful | 2 | Warm | 6 | Inefficient | 2 | Touchy | 2 | Creative | 2 |
| Energetic | 9 | Cooperative | 3 | Practical | 7 | Jealous | 2 | Intellectual | 5 |
| Quiet | 1 | Distant | 1 | Sloppy | 1 | Mellow | 2 | Ordinary | 1 |
| Shy | 9 | Cold | 1 | Disorganized | 1 | Temperamenta | 7 | Complex | 8 |
| Extraverted | 2 | Rude | 1 | Efficient | 6 | Fretful | 1 | Imaginative | 1 |

Figure 5.: Platform to collect data.

## 3.3 PROPOSED APPROACHES

Three different solutions to this problem are proposed. After developing different processes and bench-marking each one of them, by the end of this work we will be able to choose the one that best fits our needs. The following approaches are proposed: *Approach I - Multiple Independent Supervised Trait Regressors (Approach I)* on **Section 3.3.1**; *Approach II - Multiple Independent Supervised Scale Regressors (Approach II)* on **Section 3.3.2**; *Approach III - Supervised Multi-class Bin Classification (Approach III)* on **Section 3.3.3**.

### 3.3.1 *Approach I - Multiple Independent Supervised Trait Regressors*

Approach I - Multiple Independent Supervised Trait Regressors is the the first proposed approach. This uses regression techniques to calculate each traits' score. A total of five different models are trained for each trait. The input for each one of them are the selection flag of each adjective. The output of these models are the scores for each of the OCEAN dimensions. *Approach I - Multiple Independent Supervised Trait Regressors (Approach I)* has the following characteristics:

- Input is the selection/not selection of the adjectives;
- Each one of the OCEAN is considered a class;
- Each trait score is calculated independently of the others;
- The output is the probability of each of the classes;
- A total of 5 regressors (one for each trait) are calculated.

**Figure** 6 represents the operation of this approach.

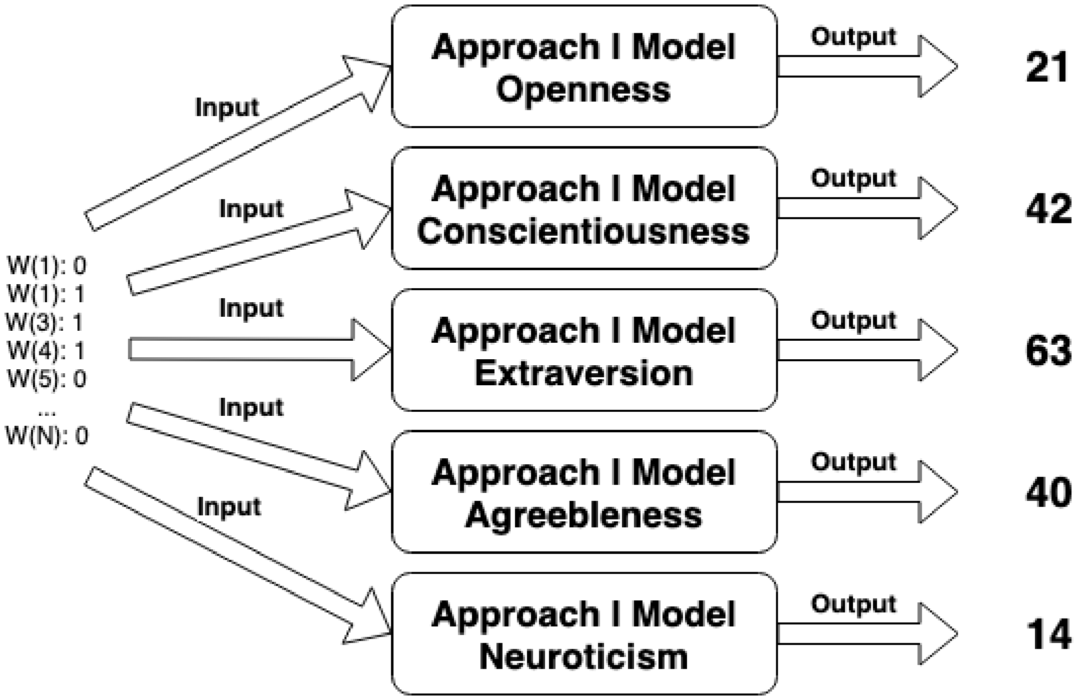

W(1): 0
W(1): 1
W(3): 1
W(4): 1
W(5): 0
...
W(N): 0
Input
Input
Input
Input
Input
Approach I Model Openness
Approach I Model Conscientiousness
Approach I Model Extraversion
Approach I Model Agreebleness
Approach I Model Neuroticism
Output
Output
Output
Output
Output
21
42
63
40
14

Figure 6.: Approach I - Multiple Independent Supervised Trait Regressors.

### 3.3.2 *Approach II - Multiple Independent Supervised Scale Regressors*

The proposed *Approach II - Multiple Independent Supervised Scale Regressors (Approach II)* uses Regression Techniques (Section 2.4.3) to calculate a proper value of the MM test scale to each of its adjectives. In this approach, multiple regression models are created, one for each adjective, each one of them is trained with selection data, whether the subject selected or not the adjectives. In the end, each model's output is the proper scale for a single word. Having all the scales of each one of the adjectives, the already existing formula from MM test is used to calculate the measurements of the *The Big Five Factors (B5F)*. Approach II has the following characteristics:

- The input for each of the models is the selection/not selection of adjectives;
- The output is the scale for each of the 40 adjectives from the MM test;
- Using the pre-established formula, B5F is measured using the calculated scales;
- A total of 40 regressors (one for each word) are calculated.

**Figure** 7 represents the operation of this approach.

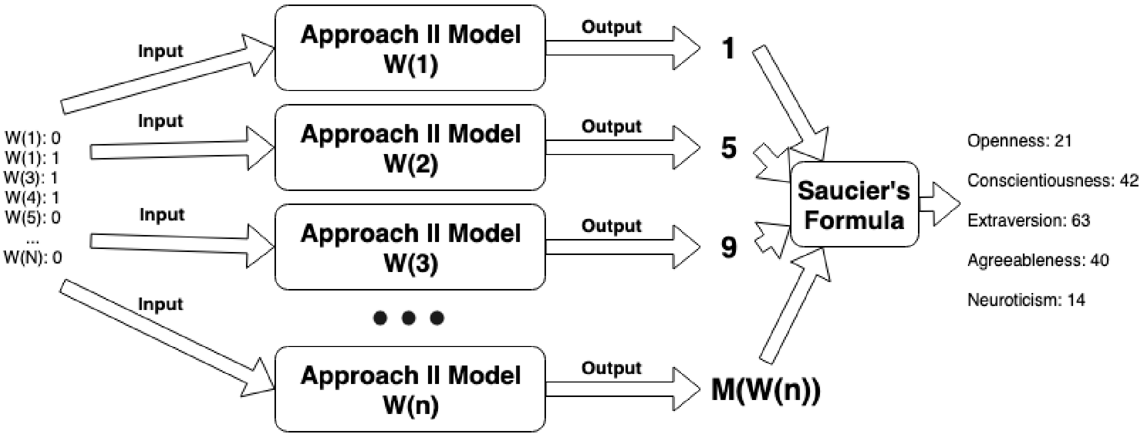


Figure 7.: Approach II - Multiple Independent Supervised Scale Regressors.

### 3.3.3 *Approach III - Supervised Multi-class Bin Classification*

The proposed *Approach III - Supervised Multi-class Bin Classification (Approach III)* uses Classification Techniques (Section 2.4.3) in order to measure each one of the B5F. This approach relies on five different models, one for each of the B5F, each one being trained with selection data, whether the subject selected or not each of the adjectives. In the end, each model's output is the class (one bin after traits' scores binarization) of a specific OCEAN's trait the subject belongs to. Approach III has the following characteristics:

- The input for each of the models is the selection/not selection of all adjectives;
- The output of each model is the class for a specific OCEAN's dimension;

- A total of 5 classification models (one for each trait) is trained.

**Figure** 8 represents the operation of this approach.

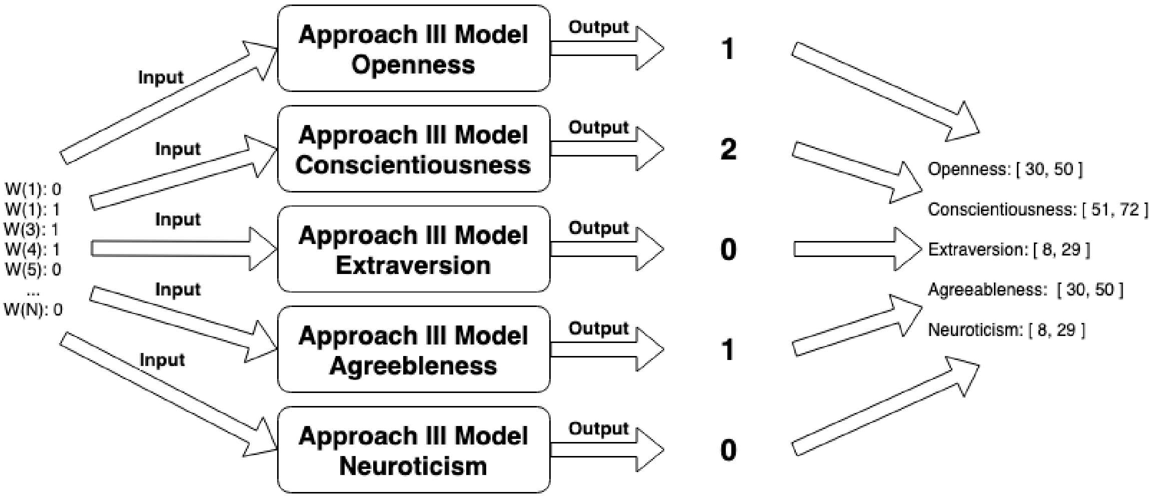

W(1): 0
W(1): 1
W(3): 1
W(4): 1
W(5): 0
...
W(N): 0
Input
Input
Input
Input
Input
Approach III Model Openness
Approach III Model Conscientiousness
Approach III Model Extraversion
Approach III Model Agreebleness
Approach III Model Neuroticism
Output
Output
Output
Output
Output
1
2
0
1
0
Openness: [ 30, 50 ]
Conscientiousness: [ 51, 72 ]
Extraversion: [ 8, 29 ]
Agreeableness: [ 30, 50 ]
Neuroticism: [ 8, 29 ]

Figure 8.: Approach III - Supervised Multi-class Bin Classification.

# 4

# MODEL DESIGN AND DEVELOPMENT

## 4.1 TECHNOLOGIES & FRAMEWORKS

Several experiments were conducted along this project. This was possible thanks to the open source libraries available in the community. Independently of the used frameworks, to develop all experiments, models, manipulate and analyse data, the useful Python open source programming language was picked, being this easy-to-read and powerful with several powerful already developed frameworks. A very important tool is *NumPy* [1], a package for scientific computing which allows to process N-dimensional data, becoming much easier to work with generic data. *NumPy* is licensed under the BSD license, enabling reuse with few restrictions. *Pandas* [2] and *Scikit-learn* [3] are the main packages to manipulate data and work with Machine Learning techniques. *Pandas* provides high-performance, easy-to-use data structures and data analysis tools. *Scikit-learn* provides simple and efficient data mining and data analysis tools built on NumPy, SciPy, and matplotlib. For data visualizations, both *Matplotlib* [4] and *Searborn* [5] packages were used. All the visualizations presented on this dissertation were produced using one of these tools. Another useful framework used for pattern mining is *Mlxtend* [6], providing usefull tools for data science tasks including Frequent Itemset Mining tecniques.

[1] Numpy.org. (2019). NumPy  NumPy. [online] Available at: https://numpy.org [Accessed 18 Sep. 2019].

[2] Pandas.pydata.org. (2019). Python Data Analysis Library  pandas: Python Data Analysis Library. [online] Available at: https://pandas.pydata.org [Accessed 18 Sep. 2019].

[3] Scikit-learn.github.io. (2019). scikit-learn: machine learning in Python  scikit-learn 0.21.3 documentation. [online] Available at: http://scikit-learn.github.io/stable [Accessed 18 Sep. 2019].

[4] Matplotlib.org. (2019). Matplotlib: Python plotting  Matplotlib 3.1.1 documentation. [online] Available at: https://matplotlib.org [Accessed 18 Sep. 2019].

[5] Seaborn.pydata.org. (2019). seaborn: statistical data visualization  seaborn 0.9.0 documentation. [online] Available at: https://seaborn.pydata.org [Accessed 18 Sep. 2019].

[6] Raschka, S. (2019). Home - mlxtend. [online] Rasbt.github.io. Available at: http://rasbt.github.io/mlxtend/ [Accessed 18 Sep. 2019].

## 4.2 VALIDATIONS TO PERFORM

As depicted on **Figure 9**, association **one** has been already proved by Goldberg's personality test [1]. A few years later, Saucier developed a reduction of Goldberg's test [2] which established relationships **two** and **three**. Barlett [8] and Kotov [11] on different studies proved the existence of the relationship **eight**, using Goldberg's Personality Test [1].

In order to ensure this work results on valid results, different tests must be performed to ensure the already established relationships remain valid. All the approaches proposed on **Section 3.3**, intend to create a Saucier MM test reduction **eight**, where the results from this new test, must be validated against the original one **five**. The even deeper proposed reduction **six**, the PP Test, brings new results, which should also be related to the original Saucier′ test **seven**.

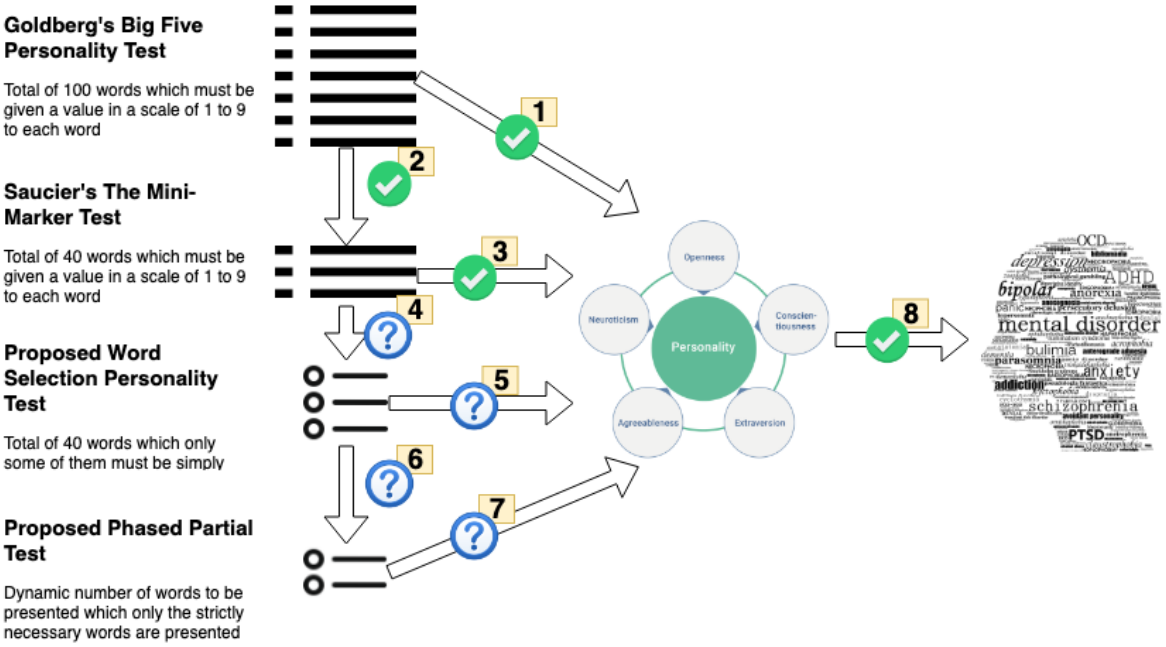

Goldberg's Big Five Personality Test
Total of 100 words which must be given a value in a scale of 1 to 9 to each word
Saucier's The Mini-Marker Test
Total of 40 words which must be given a value in a scale of 1 to 9 to each word
Proposed Word Selection Personality Test
Total of 40 words which only some of them must be simply
Proposed Phased Partial Test
Dynamic number of words to be presented which only the strictly necessary words are presented
1
2
3
4
5
6
7
8
Openness
Conscien-tiousness
Extraversion
Agreeableness
Neuroticism
Personality
OCD
depression
bipolar
ADHD
anorexia
panic
mental disorder
bulimia
parasomnia
anxiety
addiction
schizophrenia
PTSD

Figure 9.: Tests to perform in order to validate already established relationships. Legend: ”√”: Already established relationship; ”?”: Relationship to be established by this work.

## 4.3 DECISIONS

### 4.3.1 *Data collection in person*

As described on **Section 3.2**, data was initially collected using the presented platform. However, by evaluating the first **Backup Date** [7] (2019-02-26), as we can see on **Figure 10** on *blue line*, the ratio of records with selected adjectives was below 20%. Due to this, in addition to the platform, data was also collected in person. This was performed in multiple cities of Portugal: Braga, Esposende, Lisboa and Viana do Castelo.

As result, in the following Backup Dates, (2019-04-23) and (2019-05-24), the ratio as increased to around 70% and 80% respectively. This increase was accomplished because during the in-person collection it was ensured people performed the scale of each adjective but also the adjective selection.

### 4.3.2 *Automatic Selection of Words*

During the data collection process, two different methods were used: 1) the online platform described on **Section 3.2**; 2) data collection in person. As explained on **Section 4.3.1**, data collection in person was performed for several reasons, including the huge amount of records with no words selected. To make the most of data, a process of **automatic selection of words** is performed to automatically select words for those records. This auto selection is performed by using a specific threshold of scale which decides to mark a word as selected or not. This threshold is explained below.

On **Figure 10** we can see the analysis of records with and without selected words. X-axis is *Backup Date*, which represents a set of data collected until a specific date. The *blue line* represents the evolution of the records % with selected words. Both on 2019-04-23 and 2019-05-24 there is a significant increase of this %. This event is due to the in data person collection. Although there is an increase of this ratio, as already stated every single record is important to consider. The divergence of *green* (records with words selected) and *red lines* (records without words selected), which can also be represented by the *blue line*, shows the increase of incomplete records despite the increase of collected records.

Having this in mind, a auto selection of words was performed on those records. In order to have an auto-selection as reliable as possible, the scales of each adjective of each records were analysed, finding the best scale to consider a word to be selected. On **Figure 11** *yellow line* represents the ratio of selected words (y-axis) for each scale (x-axis). On the other hand, *blue line* represents the ratio of words not selected. As we can see, between scale 6 and 7, both these lines intersect each other, which should be the optimal scale to consider a

[7] Set of data collected until a specific date.

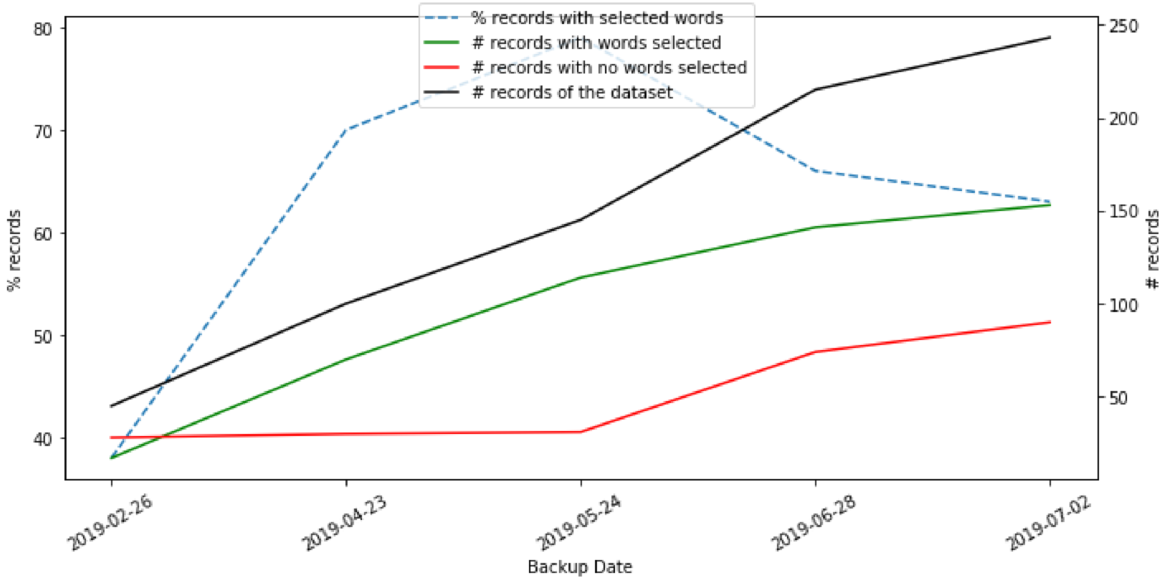


Figure 10.: Analysis of records with/without selected words.

selection over a non selection. Thus, in a simple auto-selection, all words with scale≥7, should be considered as a selection. An individual analysis of word selection on each scale for each word is available in the end of **Section 6** on **Figure 28**.

In the end of the auto-selection, every record ends with at least one selected word. Although the auto-selection method solves the uncompleted records, it presents both advantages and disadvantages. The main advantage is that every single record is used and considered, having no residuals. Besides that, the auto-selection, ensures some reliability of not having a random selection. On the other hand, as main disadvantage, this method adds noise to the data.

### 4.3.3 *Random Search over Grid Search*

Grid Search allows to build and train a model for each single possible configured hyperparameters combination. On the other hand, Random Search allows to randomly test a subset of configured hyperparameters combinations. When both are compared in the same domain, Random Search is able to converge into a better configuration within a small fraction of the computation time. This is possible because Random Search looks for configurations in a larger configuration space [22]. Independently of the modelling approach, Random Search method was used.

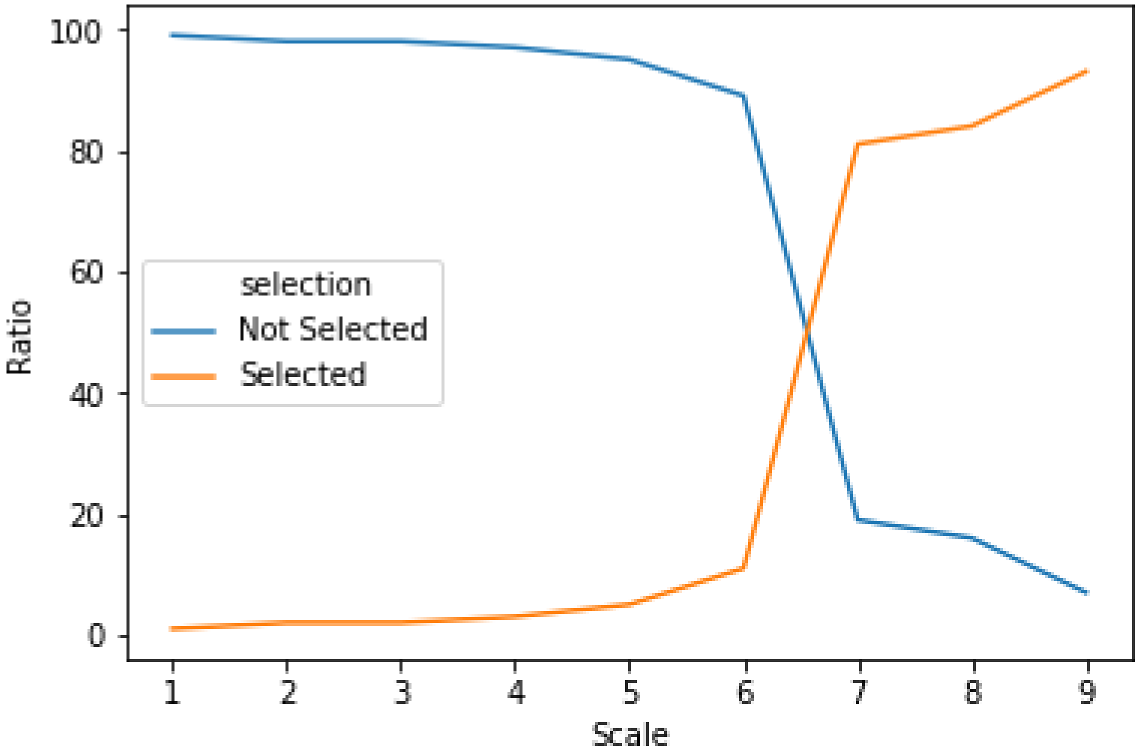


Figure 11.: Selection ratio per scale.

### 4.3.4 *k-fold Cross-Validation over Shuffle Split*

k-fold Cross-Validation divides the entire dataset into a predefined $k$ folds, in which every single sample must be in one and only one fold [8]. On the other hand, Shuffle Split, randomly samples the entire dataset during each iteration generating a training and testing set. Since data is being sample from the entire dataset on each iteration, every instance can be used multiple times for training and testing. k-fold Cross-Validation, ensures that each instance can only be used for testing one and only one time, where during each round, one fold is selected as test set and the rest as training.

By choosing k-fold Cross-Validation over Shuffle Split, independently of the modelling approach, we can ensure that out model is tested against every single instance of the original dataset leading to a more confident evaluation which is useful in cases of a limited input data.

[8] Fold is a subset of the dataset

### 4.3.5 *Root Mean Square Error over Mean Absolute Error*

*Root Mean Square Error (RMSE)* and *Mean Absolute Error (MAE)* are common metrics used to evaluate the performance of continuous regression Machine Learning models. However, it is important to understand what is the most suitable to use in the Approach I and Approach II.

MAE measures the average magnitude of the errors of a predictions' set ignoring their direction. This calculates the average of the absolute differences between predictions and the actual observations, considering every individual equally. On the other hand, RMSE a evaluation metric that measures the average magnitude of the error, calculated as the square root of the average of squared differences between predictions and the actual observations. Although both have the same property of being expressed on the same units as the variable being analysed they are distinct in some aspects.

In the case of Root Mean Square Error, since errors are squared before being averaged, it gives gives a relatively high weight to large errors. Meaning this metric should be used in situations were being off by 20 is more than twice as bad as being off by 10.

Since, in this situations, it is preferable to have smaller errors on multiple subjects than having a subjects with large errors, RMSE will be preferably used. However, both of these metrics will be considered and analysed. MAE will also be considered because it is less sensitive to outliers.

### 4.3.6 *Boosted Trees*

Several Machine Learning algorithms are available nowadays, however, each of them present advantages and disadvantages in certain conditions. These conditions can be of different nature such as: 1) data dimensions; 2) type of data; 3) data diversity and 4) data distributions. For these reasons, Boosted Trees based algorithms were preferred during modelling. XGBoost [25] is one of a long list of algorithms, having shown several optimal results on multiple studies, [26], [27] and [28].

The used data for model training has low dimensions, which invalidates usage of other good candidates such as LightGBM [29], which requires higher data dimensions to avoid overfitting. Having low dimensions of data the probability of overfitting is much higher, however, as previously referred, this implementation implements regularization natively, which helps preventing overfitting.

Since the used data is based on categorical data, more specifically of binary data, this becomes an advantage because Boosted Decision Trees work very well for this type of features, where the gain criteria works specially well for boolean features. One of the many advantages is that decision tree methods needs low amount of data transformations, allowing us to give the model data with only a few transformations. These and the previously described particularities on **Sub Section 2.4.4** shows this a good candidate for all modelling approaches.

### 4.3.7 *Number of bins on Approach III*

During Approach III - Supervised Multi-class Bin Classification binarization of the target value was required, being this performed to build a total of three bins. The number of bins was established as three for two main reasons: 1) originally, traits' values were already categorised/binarized on three different groups of values *low*, *normal* and *high*; 2) if a higher number of bins were used, there would not exist enough number of instances for each bin, representing a more difficult situation to train model. Indeed, models were tested and evaluated with a total of five bins, resulting in worst results.

### 4.3.8 *Micro vs Macro average on F1 Score*

Approach III's targets are highly imbalanced, which requires caution when choosing an evaluation metric. Both micro- and macro-averages compute different metrics, changing its interpretation. While a macro-average compute the metric independently for each each class and then computes the average, treating all classes equally, a micro-average aggregates all classes' contributions to compute the average metric. In a situation as the one of this work, a multi-class classification setup with highly imbalanced classes, micro-average is preferable since making mistakes on the less common classes is more prone to happen because we have less data to train.

## 4.4 MODELLING CONCEPTUALISATION

### 4.4.1 *Data Transformation*

As described on **Section 2.4**, after *1) Data Collection* phase it is time for *2) Data Processing and Transformation* phase. This sub section describes the implemented transformations and analysis performed during this period. Although a few transformations became irrelevant during the period of development of this project, these helped to converge into the final version of transformations to apply.

*Scales Out Boundaries*

The original test [2] required a scale within the boundaries of *1 (Extremely Not Accurate)* and *9 (Extremely Accurate)*, being this also required by the proposed solution of this thesis. However, some of the collected records did not fulfilled this requirement, whether because of a selected scale below *1* or above *9*. As we can see on **Figure 12**, more than 24% of the records have at least one adjective with a scale out of boundaries. However, these records are mainly cause by a selection of a scale below *1*, representing almost 24% of these records.

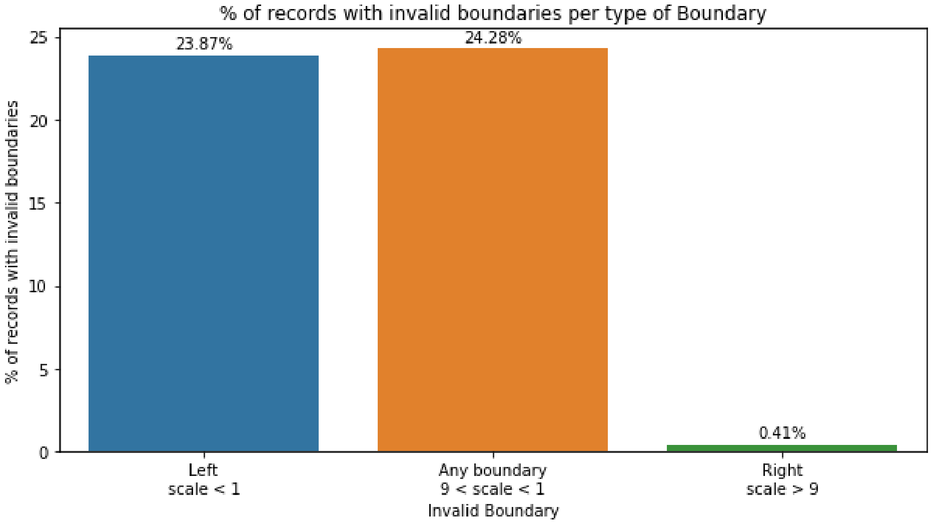


Figure 12.: Percentage of records with invalid boundaries by type of boundary.

Although it is recognised that these invalid scales are mostly represented in the lower limit, it is also important to understand if this behaviour is reflected equally in every single word. As we can see in **Figure 13** this behaviour is actually not being revealed equally in all words. We have words in both extremes, with words like *'average'* and *'touchy'* with more than *25* records, on the other side, we have words like *'talkative'* and *'sympathetic'* with less than *5* records.

In order to ensure a higher number of valid records used during modelling, to these records with invalid scales it was associated the nearest valid value. The records with invalid left boundary, the scale value *1* was used, on the other hand, records with invalid right boundary, the scale value *9* was used.

*Words Auto Selection*

After out boundaries scales correction, it still exists records with no words selected at all as described on **Sub Section 4.3.2**.

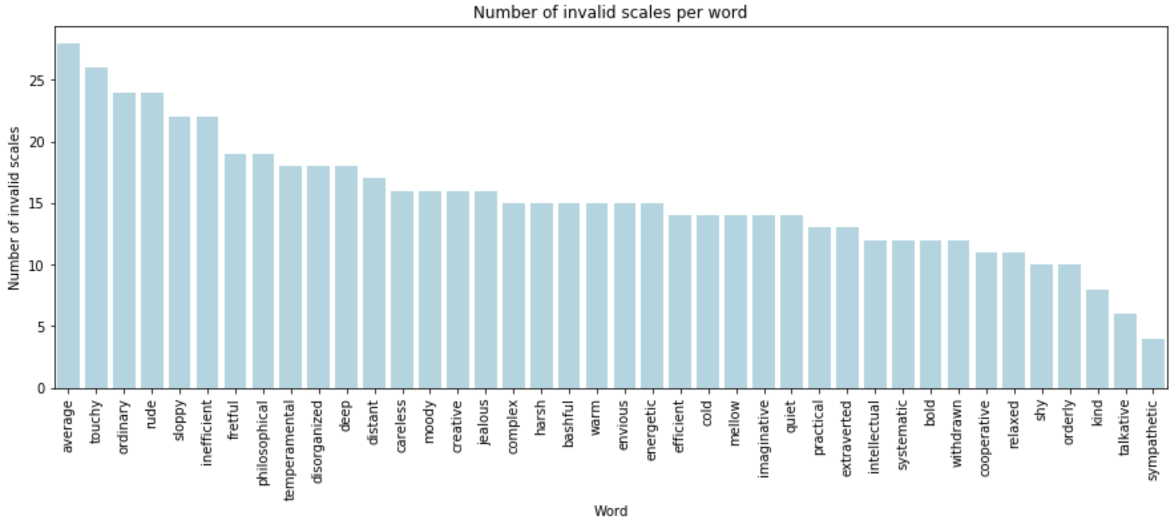


Figure 13.: Number of out boundaries scales' selections per word.

As mentioned before, the dataset size is one of the main concerns. Having this in mind, one of the data transformation operations performed was words auto selection. This operation was performed as described on **Sub Section 4.3.2**.

*One Hot Encoding*

The original dataset is persisted as *CSV* format, with the list of selected adjectives in a *1 Dimension* format. An important step is to convert this information from the original format into a *n-dimensional matrix*, which allows these information to be easily manipulated.

| index | selected_words |
|---|---|
| 0 | orderly, envious, deep |
| 1 | talkative, orderly |
| 2 | sympathetic, envious |
| 3 | deep |
| 4 | envious |

Table 1.: Original representation of words selection. Words shown on column *'selected_words'* were the ones selected by the subject.

**Table 1** shows how data is originally stored. Having all the selected words in a column precludes these data from being used for training a model. Hence, the selected words feature was one hot encoded as shown in **Table 2**, allowing this data to be given to a Machine Learning model.

| index | talkative | sympathetic | orderly | envious | deep | ... |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 0 | ... |
| 1 | 1 | 0 | 1 | 0 | 0 | ... |
| 2 | 0 | 1 | 0 | 1 | 0 | ... |
| 3 | 0 | 0 | 0 | 0 | 1 | ... |
| 4 | 0 | 0 | 0 | 1 | 0 | ... |

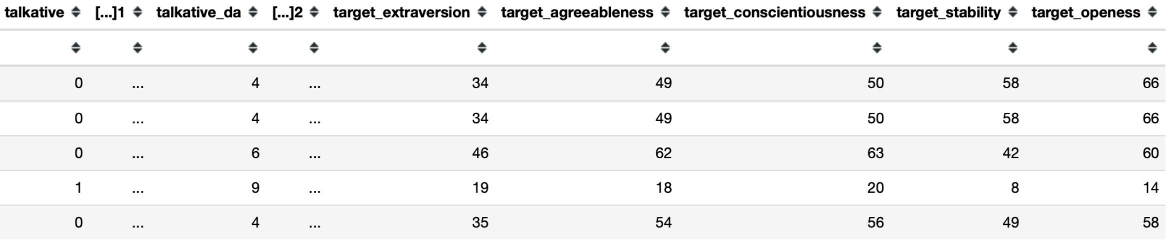
Table 2.: Words selection represented after being one hot encoded. Columns with value '1' represents selection, '0' represents a non selection.

*Transformations for Approach I*

The simplest Approach described on **Section 3.3** is Approach I - Multiple Independent Supervised Trait Regressors, being the one with less transformations required.

After the previously described transformations are applied, a single dataset is generated being this composed by two main parts required for a Supervised Machine Learning model such as Approach I, i.e., instances ($X$) and labels for each instance ($Y$).

**Figure 14** represent an example of the output of this Data Transformation process for the described Approach. The first columns of this data represents the X data (which is omitted by *'[...]1'*), having these columns the information of whether the subject selected or not the word of the respective column. The following columns, *'talkative_da'* and *'[...]2'* represents the appended auxiliary columns for the more advanced used implementations of Data Augmentation. The last part of the dataset (the set of columns *'target_extraversion'*, *'target_agreeableness'*, *'target_conscientiousness'*, *'target_stability'*, *'target_openess'*), represents the labels for each record. Although all traits are all together in the same dataset, these will be used independently on five different independent models. These columns will be used during training as labels, but also, the output of each of these five models will be compared against the respective column.



| talkative | [...]1 | talkative_da | [...]2 | target_extraversion | target_agreeableness | target_conscientiousness | target_stability | target_openess |
|---|---|---|---|---|---|---|---|---|
| 0 | ... | 4 | ... | 34 | 49 | 50 | 58 | 66 |
| 0 | ... | 4 | ... | 34 | 49 | 50 | 58 | 66 |
| 0 | ... | 6 | ... | 46 | 62 | 63 | 42 | 60 |
| 1 | ... | 9 | ... | 19 | 18 | 20 | 8 | 14 |
| 0 | ... | 4 | ... | 35 | 54 | 56 | 49 | 58 |

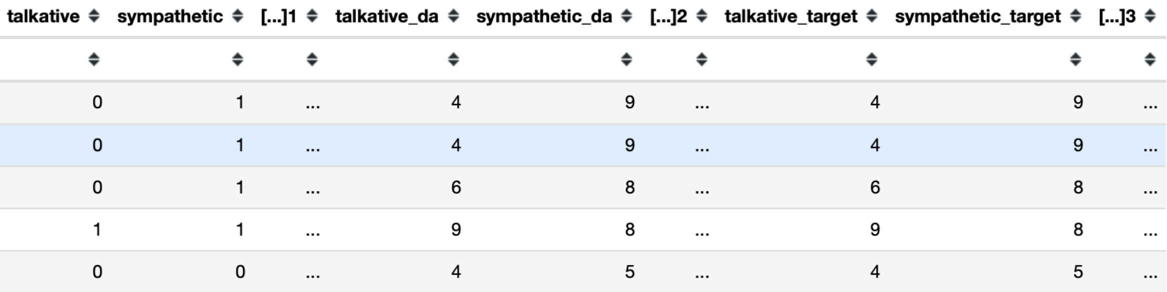
Figure 14.: Example of data ready for Approach I - Multiple Independent Supervised Trait Regressors modelling. On *'[...]1'* column are represented the remaining 39 *words* and on *'[...]2'* the respective Data Augmentation auxiliary columns described on **Section 4.4.3**. The last 5 columns represent the labels of each instance.

*Transformations for Approach II*

The Approach II - Multiple Independent Supervised Scale Regressors has the characteristic of being composed by a high number of Machine Learning models, one per word. Due to this, it becomes the approach with the dataset with the highest number of dimensions.

After previously described transformations are applied, a single dataset is generated being this composed by two main parts [9] required for a Supervised Machine Learning model such as Approach II, instances ($X$) and labels for each instance ($Y$).

**Figure 15** represent an example of the output of this Data Transformation process. The first columns of this data represents the X data (which is omitted by *'[...]1'*), having these columns the information of whether the subject selected or not the word of the respective column. The following columns, *'talkative_da'*, *'sympathetic_da'* and *'[...]2'* represent the appended auxiliary columns for the more advanced used implementations of Data Augmentation. The last set of columns, *'talkative_target'*, *'sympathetic_target'* and *'[...]3'*, represent the subjects' selected scaled for each word, which represent the labels for each record. Although all words' scale are all together in the same dataset, these will be used independently on forty (40) different independent models. These columns will be used during training as labels, but also, the output of each of these forty models will be compared against the respective column.



| talkative ⬍ | sympathetic ⬍ | [...]1 ⬍ | talkative_da ⬍ | sympathetic_da ⬍ | [...]2 ⬍ | talkative_target ⬍ | sympathetic_target ⬍ | [...]3 ⬍ |
|---|---|---|---|---|---|---|---|---|
| ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ |
| 0 | 1 | ... | 4 | 9 | ... | 4 | 9 | ... |
| 0 | 1 | ... | 4 | 9 | ... | 4 | 9 | ... |
| 0 | 1 | ... | 6 | 8 | ... | 6 | 8 | ... |
| 1 | 1 | ... | 9 | 8 | ... | 9 | 8 | ... |
| 0 | 0 | ... | 4 | 5 | ... | 4 | 5 | ... |

Figure 15.: Example of data ready for Approach II - Multiple Independent Supervised Scale Regressors modelling. On *'[...]1'* column are represented the remaining *38 words*, on *'[...]2'*the respective Data Augmentation auxiliary columns described on **Section 4.4.3** and on *'[...]3'* the labels of each instance for each word.

*Transformations for Approach III*

Approach III - Supervised Multi-class Bin Classification requires a more data transformation than the previous described approaches.

After applying the previously described transformation operations, a couple more operations are required. For each trait to predict, it is required to binarize the values of all five traits. For this process, it is required to determine what are the limits of each bin. For this

[9] With the implementation of new Data Augmentation methods, an auxiliary third part was added

process, a set of already used bins in the original study [2] were used. These are characterised by the following values and notation: *low*: [8, 29]; *normal*: [30, 50]; *high*: [51, 72]. **Figure 16** shows the distribution of records' percentage of each bin for each trait which allows to understand that these are highly unbalanced. Every single trait presents one bin, *'normal'*, with much higher number of records. The second bin with more records is *'high'*, except for *'extraversion'* trait, and the one with lower amount of records *'low'*. These differences of values per bin, must have to be considered during modelling and training the models.

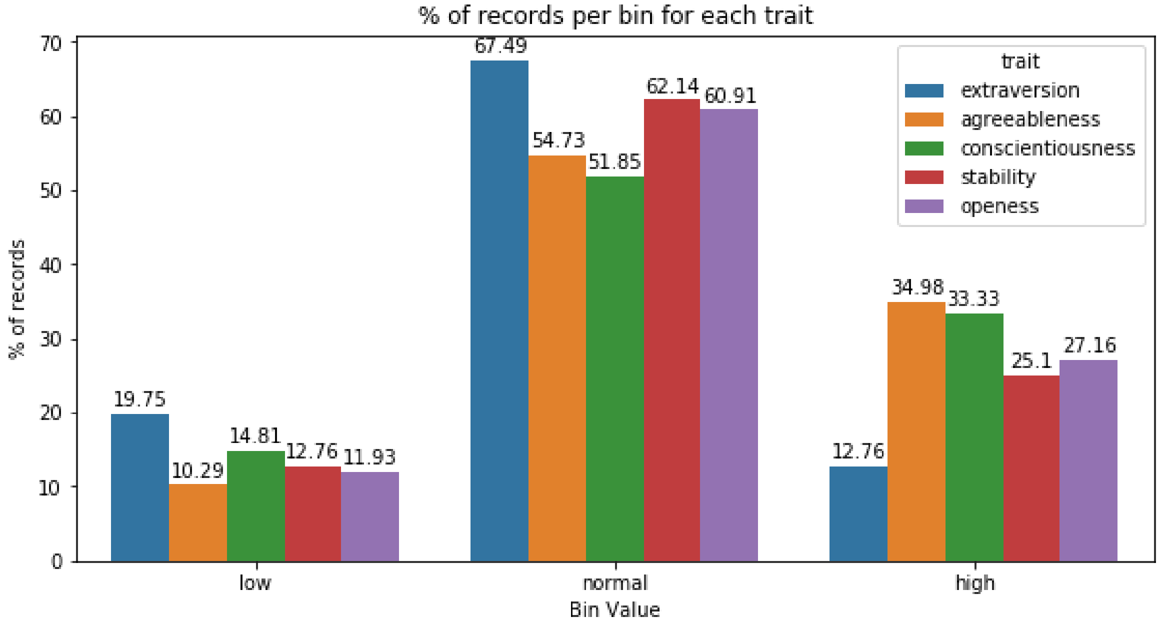


Figure 16.: Distribution of records' percentage of each bin for each trait.

However, a binarization of five bins was also experimented. The same distribution analysis is presented on **Figure 17**. These are characterised by the following values and notation: *very low*: [8, 18]; *low*: [19, 29]; *lower normal*: [30, 39]; *higher normal*: [40, 50]; *high*: [51, 61]; *very high*: [62, 72]. With this binarization configuration, these become even more unbalanced.

The final generated dataset is composed by two main parts required for a Supervised Machine Learning model such as Approach III, instances ($X$) and labels for each instance ($Y$).

**Figure 18** represent an example of the output of this Data Transformation process. The first columns of data represent the X data (which is omitted by *'[...]1'*), having these columns the information of whether the subject selected or not the word of the respective column. The following columns, *'talkative_da'* and *'[...]2'* represent the appended auxiliary columns for the more advanced used implementations of Data Augmentation. The last part of the dataset (the set of columns *'extraversion_target'*, *'agreeableness_target'*, *'conscientiousness_target'*,

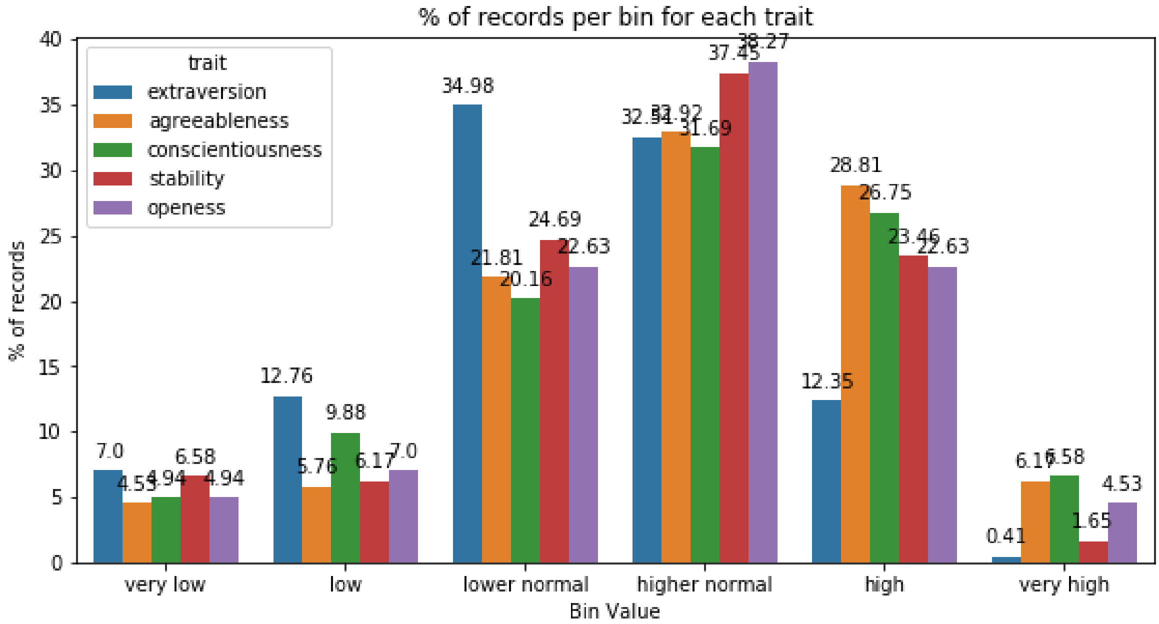
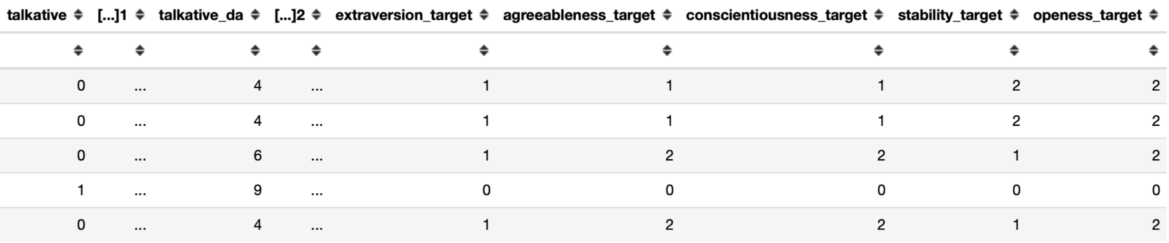


Figure 17.: Distribution of records' percentage of each bin for each trait with a binarization of 5 bins.

*'stability_target'*, *'openess_target'*), represents the labels for each record. Although all traits are together in the same dataset, these will be used independently on five different independent models. These columns will be used during training as labels, but also the output of each of these five models will be compared against the respective column.



| talkative ⬍ | [...]1 ⬍ | talkative_da ⬍ | [...]2 ⬍ | extraversion_target ⬍ | agreeableness_target ⬍ | conscientiousness_target ⬍ | stability_target ⬍ | openess_target ⬍ |
|---|---|---|---|---|---|---|---|---|
| ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ | ⬍ |
| 0 | ... | 4 | ... | 1 | 1 | 1 | 2 | 2 |
| 0 | ... | 4 | ... | 1 | 1 | 1 | 2 | 2 |
| 0 | ... | 6 | ... | 1 | 2 | 2 | 1 | 2 |
| 1 | ... | 9 | ... | 0 | 0 | 0 | 0 | 0 |
| 0 | ... | 4 | ... | 1 | 2 | 2 | 1 | 2 |

Figure 18.: Example of data ready for Approach III - Supervised Multi-class Bin Classification modelling. On *'[...]1'* column are represented the remaining 39 *words* and on *'[...]2'* the respective Data Augmentation auxiliary columns described on **Section 4.4.3**. The last 5 columns represent the labels of each instance.

### 4.4.2 *Patterns Mining*

In the process of *Patterns Mining*, a very useful type of process is the *Frequent Itemset Mining (FIM)*. By performing FIM a set of interesting associations and correlations between words were found. In this particular application APRIORI algorithm [10] was used to analyse the list of selected words of each instance resulting in a total of 6129 Frequent Itemset Mining stored in CSV format so these can be used and explored during the process.

**Table 3** shows the top 5 found by Frequent Itemset Mining with highest support but with lower length, since an item set with less elements has a higher probability of happening more. These results shows that people who select the word *"kind"* commonly select one of the following: 1) *"imaginative"*; 2) *"practical"*; 3) *"intellectual"*; 4) *"cooperative"*. On *index* 1, it shows *"creative"* and *"imaginative"* are simultaneously selected.

On the other hand, **Table 4** shows the top 5 found by Frequent Itemset Mining with lower support and so, less relevant. These results shows more complex item sets, although *"kind"* is still a very common word to be selected.

Both on **Table 3** and **Table 4** most of the commonly selected words have positive connotation such as *"kind"*, *"warm"*, *"efficient"*, *"cooperative"* and others.

| index | support | itemset | length |
|---|---|---|---|
| 0 | 31 | kind, imaginative | 2 |
| 1 | 28 | creative, imaginative | 2 |
| 2 | 27 | kind, practical | 2 |
| 3 | 27 | kind, intellectual | 2 |
| 4 | 27 | kind, cooperative | 2 |

Table 3.: Frequent Itemset Mining results - top 5 with highest support.

| index | support | itemset | length |
|---|---|---|---|
| 6125 | 5 | kind, warm, complex, philosophical | 4 |
| 6126 | 5 | kind, warm, philosophical, efficient | 4 |
| 6127 | 5 | shy, creative, philosophical | 3 |
| 6128 | 5 | kind, cooperative, philosophical, efficient | 4 |
| 6129 | 5 | kind, bold, efficient, warm, complex, imaginative, intellectual | 7 |

Table 4.: Frequent Itemset Mining results - top 5 with lower support.

In a total of 6129 found patterns, as expected, there are particular words that appear more frequently as elements of patterns. Represented on **Figure 19** is the total of found patterns per word, as can be seen, there is a big difference on the relevance of each word when it comes to patterns. While the word 'imaginative' has almost 250 found patterns, 'ordinary' has around 10 associated patterns. One aspect to notice, most of the less relevant words in this figure are associated with a negative connotation, and it is also to notice that these

[10] Implementation of Mlxtend. *"Mlxtend (machine learning extensions) is a Python library of useful tools for the day-to-day data science tasks."*

words have a lower selection frequency as represented on **Figure 20** which leads to a low amount of found patterns.

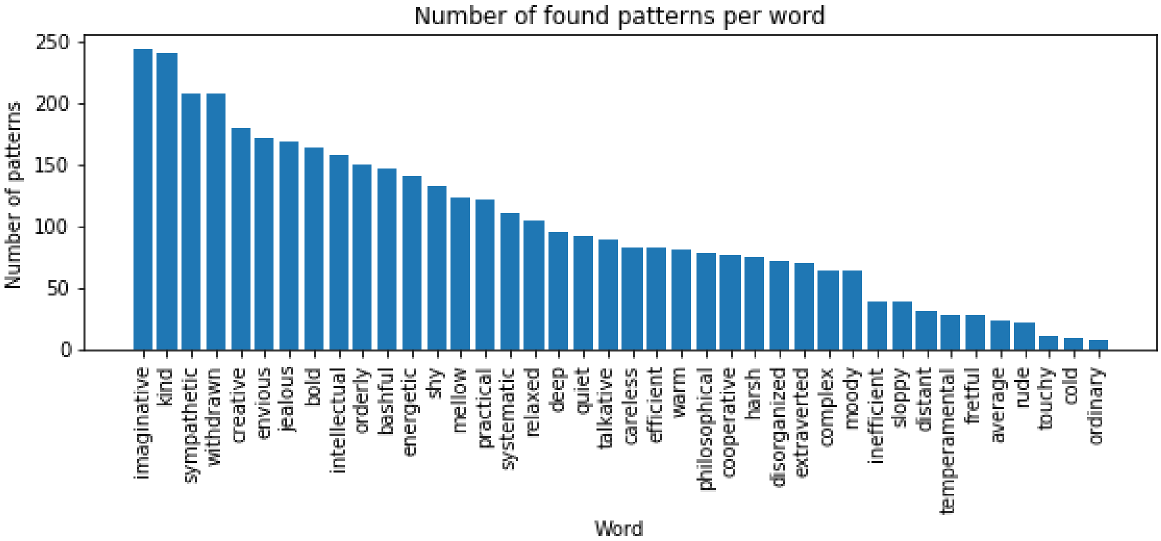


Figure 19.: Total of found patterns per word.

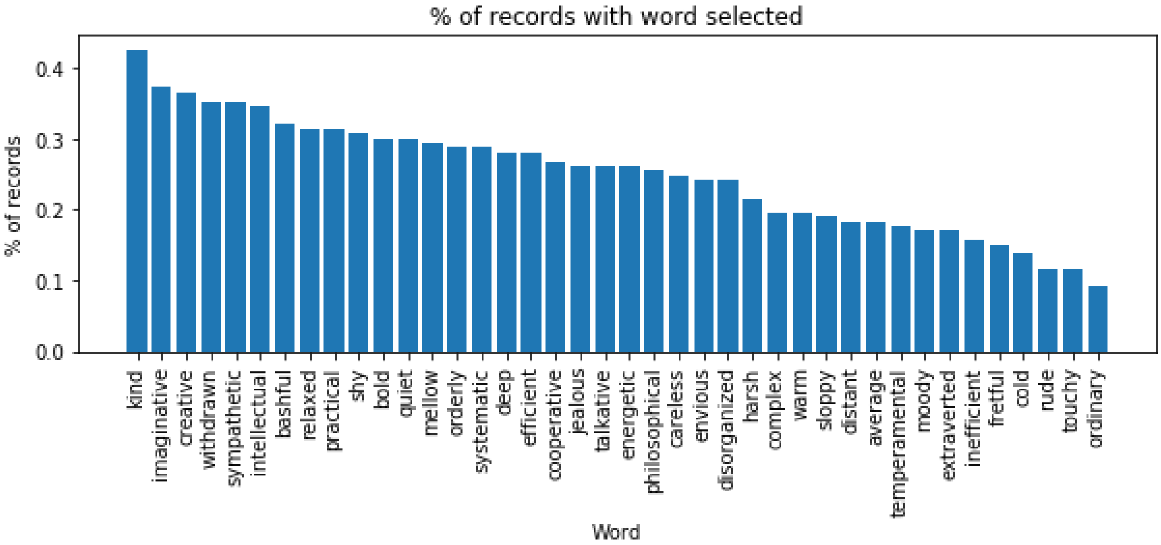


Figure 20.: Percentage of records with word selected.

The result of this analysis, Frequent Itemset Mining, it is very relevant as input for the methods explained in the next sections. With the definition of patterns, we can now prove that some words have a higher probability of being selected if $N$ other words are also selected. For instance, **Figure 21** shows the percentage of found patterns with word ’creative’ where each of the other words are also selected, with this, we understand that words like ’imaginative’ or ’shy’ are more prone to be selected by a person who selects ’creative’ than selecting 'cold' or 'touchy'. Besides the fact this information can help us to better understand our data, it also, as described in the next subsection, enable us to mimic a real life behaviour to generate new synthetic data.

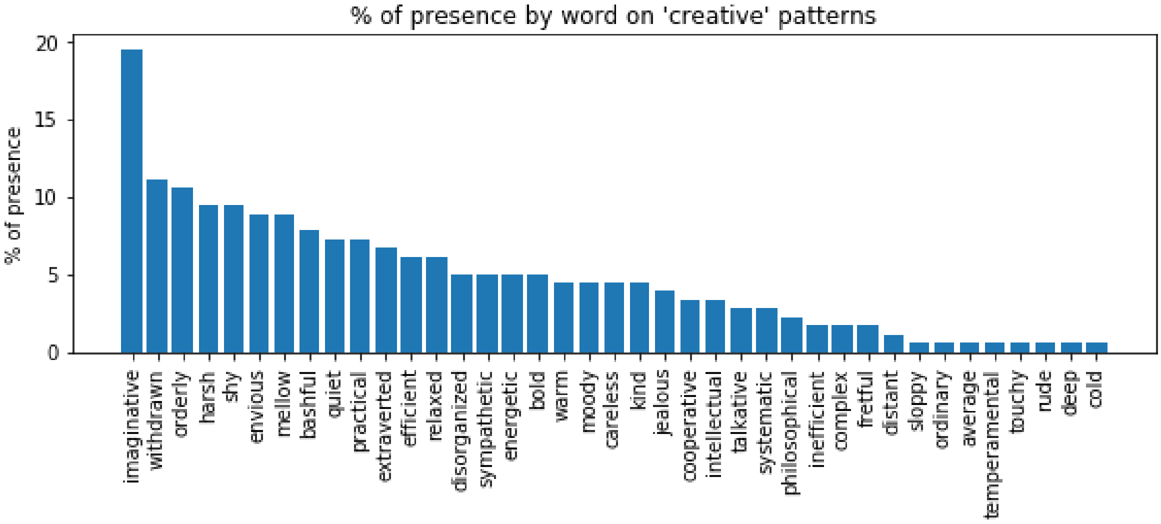


Figure 21.: Percentage of presence of each word on found patterns with word 'creative'.

All found patterns resulting from this process and with support $\geq 10$ are attached in the final **Sub Section B.3**.

### 4.4.3 *Data Augmentation Techniques*

As described previously, one of the problems faced during this work, was the small amount of available data to analyse, explore and train Machine Learning models. An existing technique used in order to reduce the effects of low amounts of data is a so called process of *Data Augmentation (DA)*. Although DA may be a well known technique, this has no list of steps to be applied in every situation, therefore, three different Data Augmentation approaches were tested and evaluated. Each approach is some sort of evaluation and improvement compared to the the previous one, being these developed iteratively, where each iteration intends to solve the issues found by the previous one.

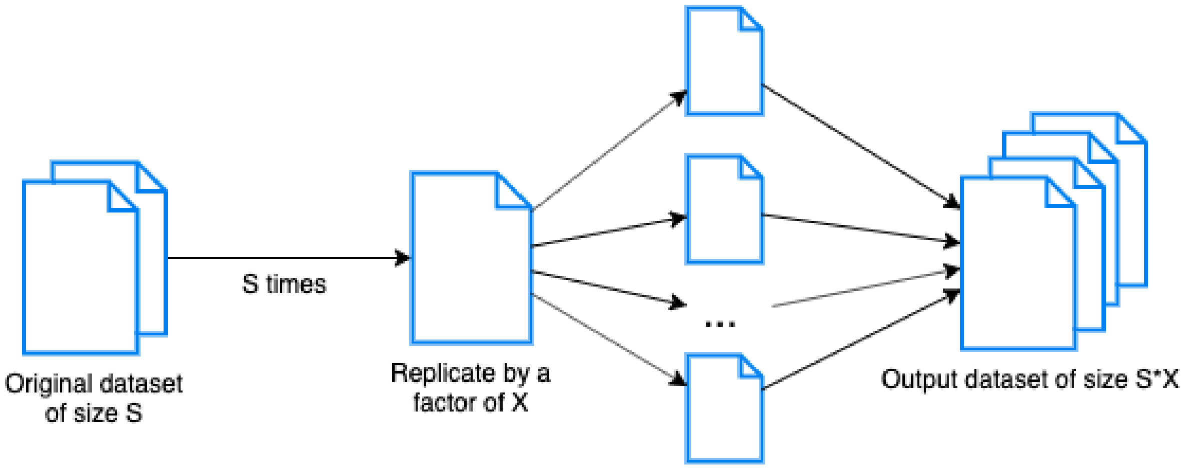
### 4.4.4 *Data Augmentation I - Simple but highly predisposed to errors*

One of the biggest advantages of this Data Augmentation method is its simplicity. Essentially, it has the entire dataset of size S as input and a new dataset output of size $S*X$, where $X$, although $X_i$=40, can be any arbitrary number to match the factor of multiplication we want to accomplish. The entire process is presented in **Figure 22**. Thus, for each row of the original dataset, X variations are generated, in which, one and only one word/adjective is varied (selected/deselected), leading to a max of 40 variations per instance.

For each word's selection or deselection, depending on the approaches described on **Section 3.3**, it may require different adjustments. Due to the nature of this approach, it only requires adjustments to the selected scale of the related word. It was given a default value for a selection and deselection, *7* and *3* respectively. These values were chosen because as we can see on **Figure 11**, *7* is the turning point where the majority of the inquiries selected a specific adjective. Concerning to the deselection value, *3*, this is at the same distance of the minimum value (*1*) as *7* is from the maximum value (*9*). In the end of this pipeline, the output will contain: 1) all instances of the original dataset; 2) X variations of each instance of the original dataset.

The described approach can be characterised by its simplicity. Fixed variations of each instance, fixed scales, one and only word is varied. Due to these properties, this method is faster to process being performed almost immediately. Another advantage, is that the final output is expected and controlled.

However, it also has several disadvantages. In case a bigger generated dataset is requested, this approach is limited to a maximum of a 40x dataset bigger than the original, since only one and only one word is changed in each new instance and there is a total of


Original dataset of size S
S times
Replicate by a factor of X
...
Output dataset of size S*X

```
if W is selected:
    W = not selected & Scale(W) = 3
else:
    W = selected & Scale(W) = 7
```

Figure 22.: Data Augmentation I - process overview. Legend: ”W”: word to variate on each replica.

40 words. Another disadvantage, is that by using *default values (7; 3)* for selection and deselection, bias is being introduced in the final dataset, where bias will increase with a greater size of the original dataset or the factor of multiplication, *X*, because these default values does not describe the reality. Therefore, after testing this approach and check the viability of Data Augmentation methods, the final method to consider, was reconsidered, resulting on a second, but also not final, iteration described on the next subsection.

### 4.4.5 *Data Augmentation II - Dynamic data augmentation*

One of the main characteristics of the implementation described on 4.4.4, is that all that the entire augmentation process is performed *prior* to training and k-fold Cross-Validation. Thus, the main breakthrough of this new revised implementation, it is a *dynamic factor*, this can be accomplished by implementing DA during each iteration of the k-fold Cross-Validation process.

Inserting a *dynamic factor* to the DA process, addresses two main advantages: 1) all data can be used both to train and evaluate the model. During each iteration of *k-fold Cross-Validation (CV)*, a new dataset is generated based on fold and the model is tested against the rest of the original dataset. Thus, there is no need to split our data into train and test sets, being an advantage given the small size of the available data. 2) unlike 4.4.4, where the augmented data is also used to test the model, this allows to test our model only against the original data, not against generated data based on a subset of data used also for training, eliminating tendencies of overfitting.

Besides the *dynamic factor*, other modifications were implemented during this iteration. As described in the previous subsection, the maximum size of the final generated dataset was listed as one of the disadvantages, this limitation was reduced by performing two main changes. While in the previous method, one and only one word was varied to generate each replica, now several words can be changed. The selection of words to variate during each replica is selected randomly where any word has the same probability to be picked. Another implemented modification are *dynamic scales*, meaning that unlike what was performed previously, using default values of 7 and 3 for selection and deselection respectively, random values are being used. Here, the scale attributed for a deselection is any random number of [1, 6], where any scale as the same probability to be selected. However, the selection value it is the same default value of 7.

With correct use of this new method, since this should be implemented during each single iteration of k-fold Cross-Validation, *k* outputs will be generated, where *k* is the number of folds used during this evaluation. As shown in **Figure 23**, this method should be called during each iteration using the training folds as input, which will return a output of a dataset both with the original training folds and the new generated data.

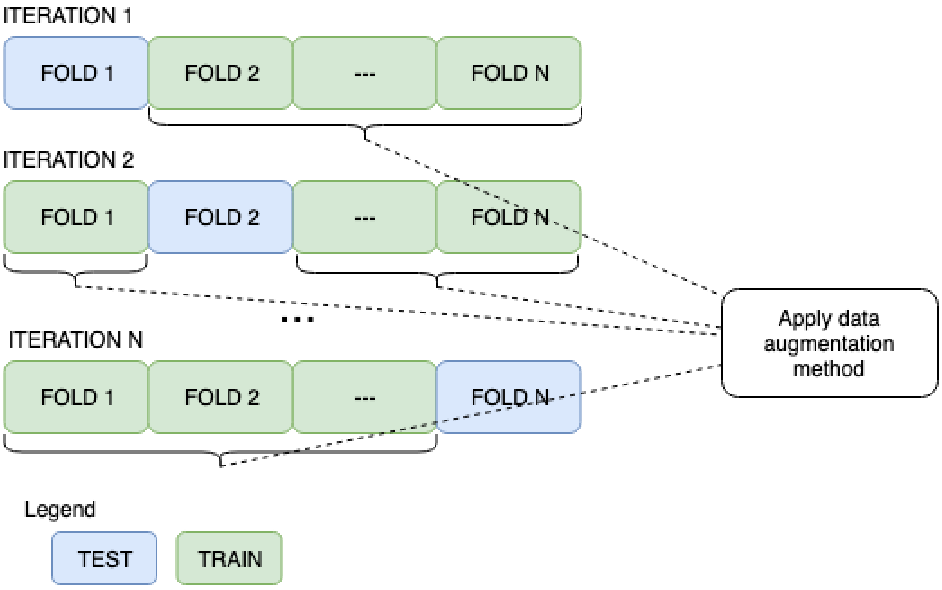


Figure 23.: Data Augmentation II - method usage. Legend: ”n”: number of folds of k-fold Cross-Validation, same as *k*;

This iteration can be described by the new dynamic implementation but also the randomness of both selected words and scales. As main advantages, we can now make use of as much data as possible both for train and test our models. The breaking of the max size of the generated data allow us to generate as much data as possible. The usage of random scales gives us more real data eliminating some of the bias previously introduced by the usage of default values. Another advantage, is that by creating the test fold before Data Augmentation, we ensure that our model is being evaluated against real data, instances that were present in the original dataset.

Although the new implementation can be considered an improvement, there are still a few problems. The number of words selected for each replica is a fixed number which will be the same number for every single replica. Limiting not only on the variety of data we can generate but also how natural our synthetic data can be. Another issue is the selection of words is completely random, representing no similarities to reality, where a word $X$ has an higher probability of being selected with $Y$ than with $W$, according to the previous subsection 4.4.2.

The described problems leaves us with room to improve the current Data Augmentation method, leading us to a new iteration.

### 4.4.6 *Data Augmentation III - Natural distributions*

In the previous Data Augmentation iteration, although some sort of randomness was addressed, it is not related to real life data which this new implementation intends to solve. This can mainly be done by replacing random processes where every single event has the same probability than the rest.

One of the first changes, although simple, in order to improve similarities with real data, besides several random words can be changed in each replica the number of words can also be different in each replica, where the boundaries of how many words are changed can be easily modified since it is parameterized. The number of words to change is a random number whitin the predefined boundaries where all numbers have the same probability.

After randomly selecting N words, using previously described instructions, there is a an entire process represented on **Figure 24** to execute for each word. For each word, it begins by deciding if a previously found pattern (**Section 4.4.2**) associated to that word must be used. Although the probability of each event is parameterized, initially values used are *P(use pattern)=0.6* and *P(not use patterns)=0.4*.

If no pattern it is used for that specific word, the next step is to set the selection flag for that word in the replica instance. This is accomplished by using the word's natural rate of selection, **Equation 1**, where if *NR(w)=.35*, the probability of that word being selected in the replica is of 35%. With this, we ensure that the output of Data Augmentation has the same distribution has the original dataset, minimising any bias previously being added.

$$NR(w) = \frac{SR(w)}{TR} \tag{1}$$

where:

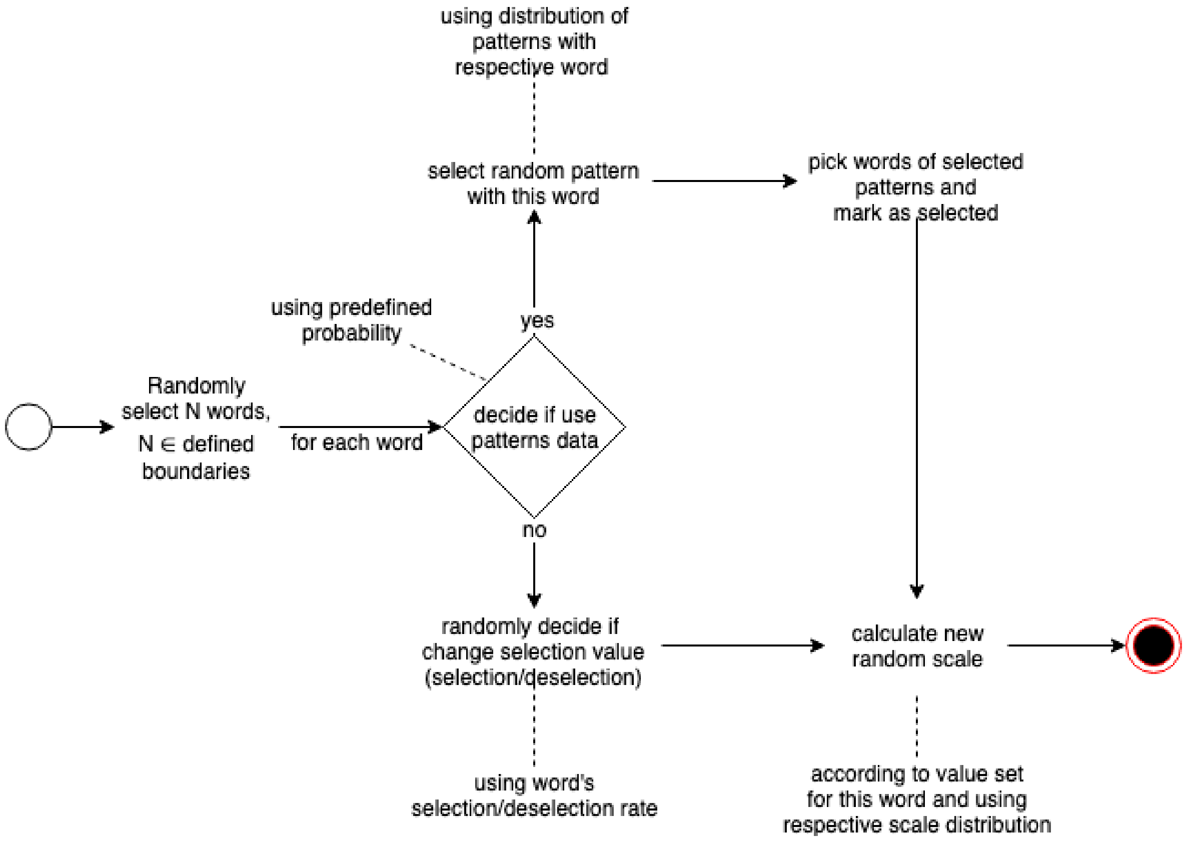

using distribution of patterns with respective word
select random pattern with this word
pick words of selected patterns and mark as selected
using predefined probability
yes
Randomly select N words, N ∈ defined boundaries
for each word
decide if use patterns data
no
randomly decide if change selection value (selection/deselection)
calculate new random scale
using word's selection/deselection rate
according to value set for this word and using respective scale distribution

Figure 24.: Data Augmentation III - process overview to execute per replica.

$NR$ = natural selection rate
$w$ = word to evaluate
$SR$ = total instances with the word selected in the original dataset
$TR$ = total number of instances in the original dataset

Regardless of previous result (selection or non-selection), a new scale is associated. This new scale is randomly selected using word's natural scale distribution, where the probability of each scale is dependent whether the word is selected or not in replica. An example of this distributions is represented in **Figure 25**. For example, if word *'creative'* is selected, the only scales that could be associated are *[4, 9]*, existing a much higher probability of being used *[7, 9]* than *[4, 6]*. This natural distributions are calculated using **Equation 2**. Once more, by using this natural distribution we ensure that the output of Data Augmentation has the same distribution has the original dataset, minimising any bias previously being added. An individual analysis of these distributions grouped by trait for each word is also presented, i.e., below in **Figure 26** *Extraversion* is analysed. The individual analysis of the remaining traits is attached on **Attachment B.1** in the following figures: *Agreeableness* - **Figure 38**; *Conscientiousness* - **Figure 39**; *Stability* - **Figure 40**; *Openness* - **Figure 41**;

$$NS(w(f)) = \frac{TR(w(f)(s))}{TR(w(f))} \tag{2}$$

where:

$NR$ = natural scale distribution
$w$ = word to evaluate
$f$ = flag of word's selection
$s$ = scale associated to word, where $s \in [1, 9]$
$TR$ = total instances
$TR(w(f)(s))$ = total instances where word $w$ has selection flag $f$ and scale $s$
$TR(w(f))$ = total instances where word $w$ has flag $f$

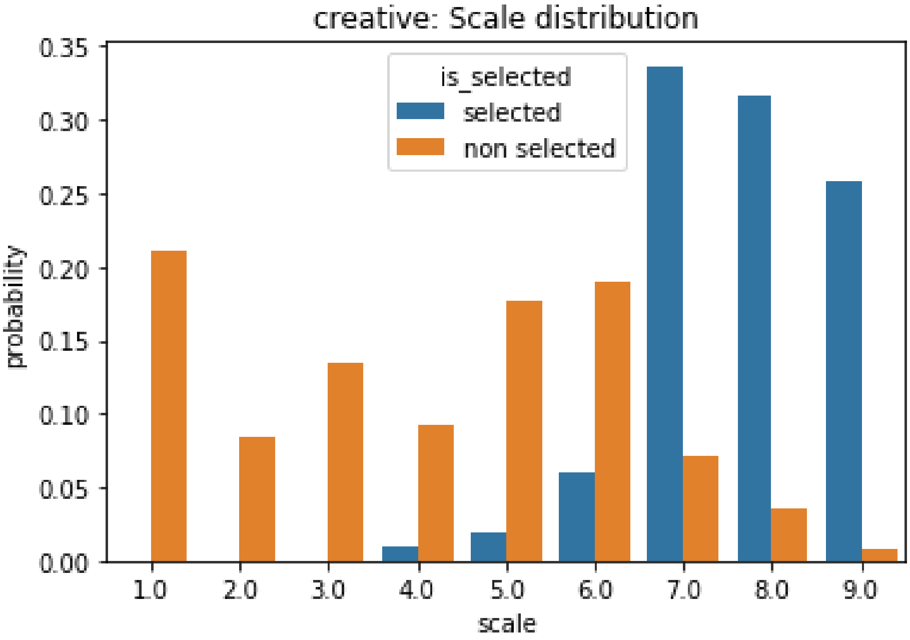


Figure 25.: 'creative' word: Scale distribution example by selection/deselection

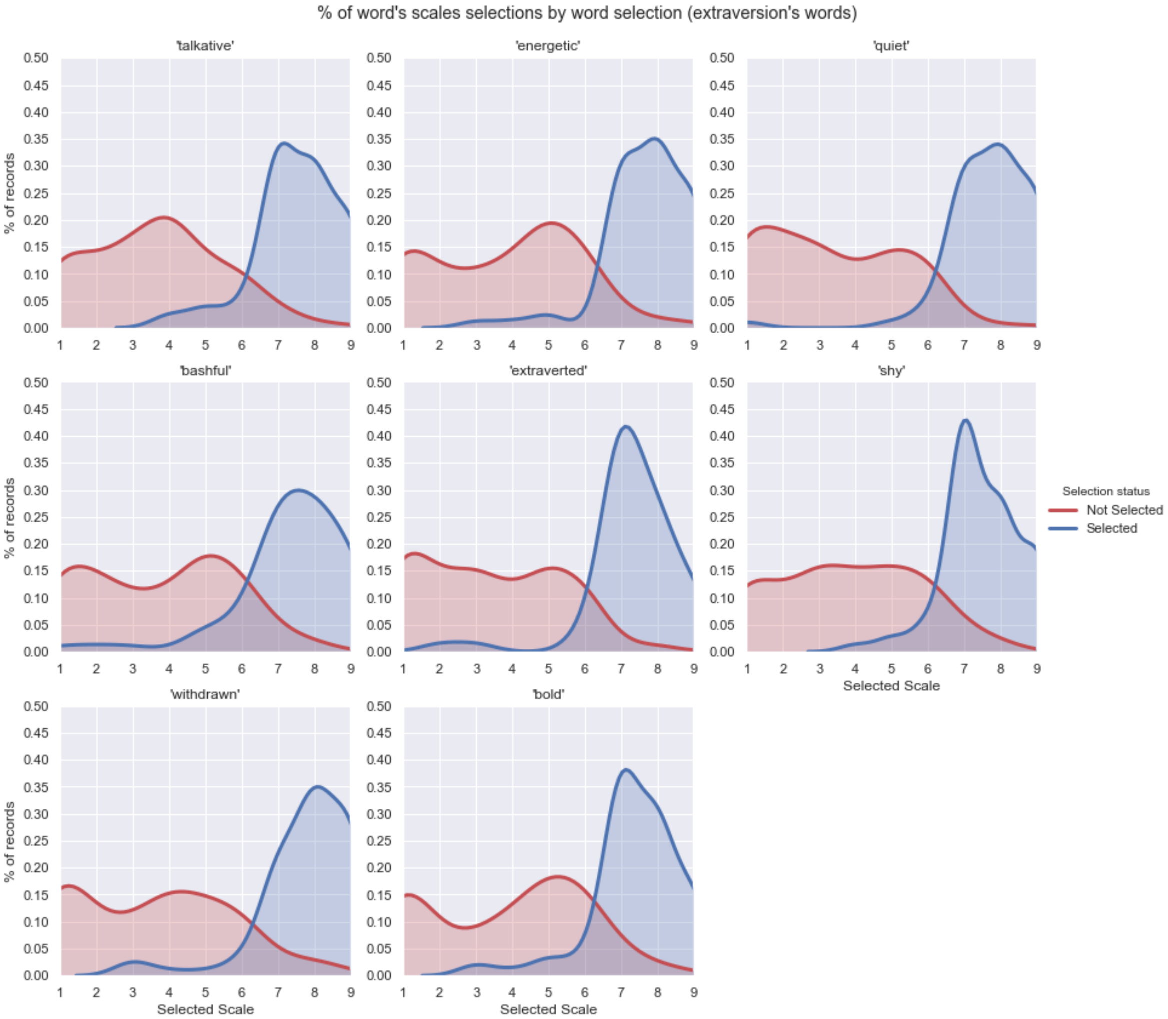


Figure 26.: Extraversion % of word's scales selections by word selection

However, if a pattern is used, the first step is to select a pattern with the respective word among all found patterns on **Section 4.4.2**. A pattern is randomly selected using the natural distribution of found patterns. This probability for each pattern is calculated by **Equation 3**, where word's *support* is divided by sum of all words' *supports*. However, since only patterns with the respective word can be selected, these probabilities are adjusted to a population with only the valid patterns. An example of this adjustment is represented on **Figure 28**. By using this natural distribution we ensure that, respecting to patterns presence, the output of Data Augmentation has the same distribution has the original dataset, minimising any bias previously being added. After selecting the pattern to be used, every word that composes the pattern is selected and used the selection flag as *selected*, this way we ensure the replica will present the pattern as well. The final step is to set a new scale for each of these words, for this, the method described above is used, using the distribution represented **Figure 27** (to notice that in the figure only the top 50 with highest weight are listed) to choose the new scales, where the weight(probability) of each pattern is calculated by **Equation 3**, this way, the natural distribution of the events is ensured, minimising noise.

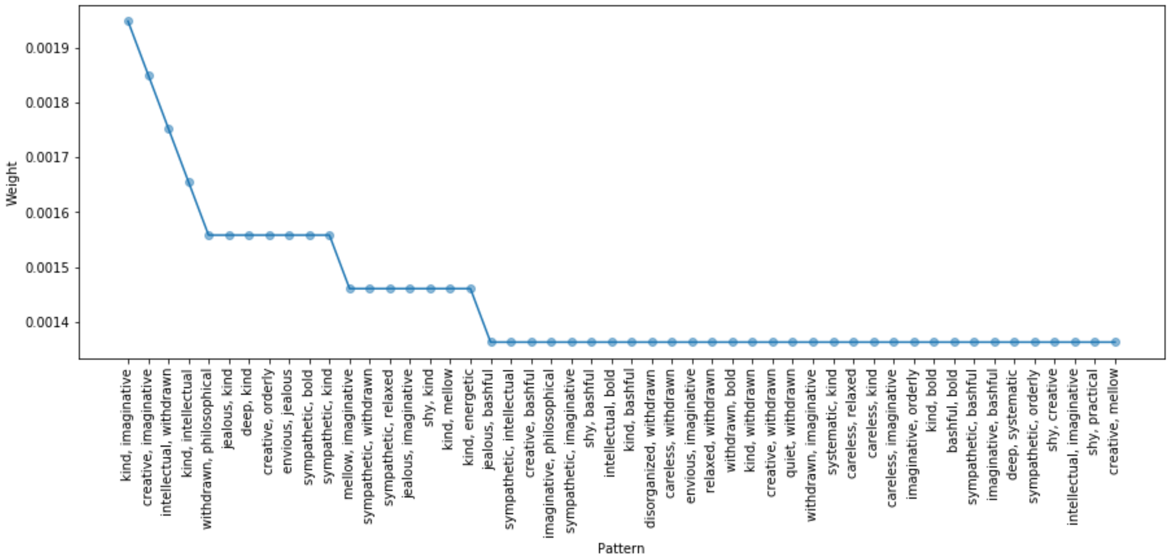


Figure 27.: Top 50 found patterns with highest relevance

$$W(p_x) = P(p_x) = \frac{support(p_x)}{\sum_{i=0}^{b} support(p_i)} \tag{3}$$

where:

$W(p_x)$ = weight of a pattern $p_x$
$P(p_x)$ = probability of a pattern $p_x$
$p_x$ = a specific pattern found on **Section 4.4.2**
$support(p_x)$ = calculated support of word $p_x$

$b$ = total number of found patterns on **Section** 4.4.2

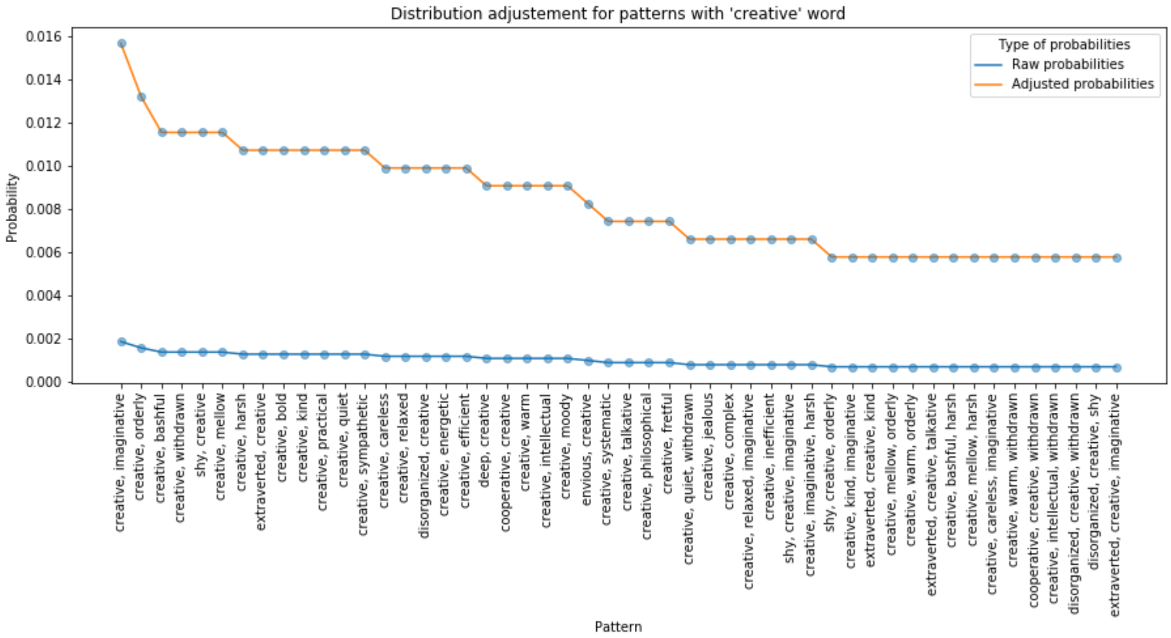


Figure 28.: Top 50 found patterns with word 'creative'

As described above, one of the mains advantages of this method is that the Data Augmentation process' output will show the same distributions as the original dataset, minimising any type of noise or bias during training models. This final iteration was used as the final tool for modelling.

### 4.4.7 *Adjustments required during modelling after Data Augmentation*

Although Data Augmentation can be considered as an isolated process, in order to implement this method to the previously developed pipeline, some adjustments were required.

Initially, the original dataset was splitted into training and testing sets right in the beginning of the entire process, before hyper parameter tuning, analysis and model training. However, both Data Augmentation methods described on 4.4.5 and 4.4.6, requires to not previously split the original dataset and perform this split during k-fold Cross-Validation, being the first required update. In order to smooth DA operation, a set of auxiliary columns were appended into the modelling data, being this columns the main input to the operation, independently of the used modelling approach (3.3). These columns contained nothing more then the original scale given to each instance, being these removed in the end of data generation.

Given the nature of Approach III described on 3.3.3, after generating new data, a set of pre-processing transformations are required. The operation to run is the binning processing.

### 4.4.8 *Modelling and Evaluation*

As described on **Section 2.4**, after both phases of *1) Data Collection* and *2) Data Processing and Transformation* are completed it is time for *3) Modelling and Training* and *4) Model Evaluation* phases. This sub section describes the designed solution for this problem.

#### *Hyperparameter Optimization*

In order to reach a good, finished and ready model, one recommended step is *Hyperparameter Optimization (Hyperparameter Tunning)*, for this purpose, python's library Scikit-learn [11] was used to fully automate the entire process of Hyperparameter Tunning, applying *RandomizedSearchCV*. This technique allows to easily apply Random Search for Hyperparameter Optimization [22] which searches for the most suited hyper parameters for the final model however, by applying k-fold Cross-Validation it also minimises hyperparameters' overfitting by picking the most suited configuration considering multiple folds of data, training a model with a certain configuration $k$ times on$k$ different training sets and validating on $k$ different validations sets.

#### *Model Training*

k-fold Cross-Validation was used to minimize overfitting during Hyperparameter Optimization, however, this should also be used during training, avoiding overfitting of the model itself. This technique can be used to evaluate the performance of our final model considering different sets for training and validation.

Initially, python's library Scikit-learn [12] was used to fully automate the entire process of k-fold Cross-Validation training, however, the supported native high level methods presented some flaws when a more customised behaviour was required, when the application of DA method described on **Section 4.4.5** was needed. In a second and final iteration of k-fold Cross-Validation, although *Scikit-learn* was also used, a more lower level API was used. In each iteration of CV, although each Approach described on **Section 3.3** has their own peculiarities, the behaviour is the same:

- Split the original dataset into training and validation according to iteration's folds;

[11] Scikit-learn.org. (2019). scikit-learn: machine learning in Python scikit-learn 0.21.3 documentation. [online] Available at: https://scikit-learn.org/stable/ [Accessed 26 Aug. 2019].

[12] Scikit-learn.org. (2019). scikit-learn: machine learning in Python scikit-learn 0.21.3 documentation. [online] Available at: https://scikit-learn.org/stable/ [Accessed 26 Aug. 2019].

- Apply Data Augmentation if needed;
- Apply data transformations and clean residual data;
- Use training folds for model training;
- Use trained model to predict test folds;
- Compare test's predictions to real test's values.

In the end of all iterations the *median* of all models' results are calculated, being this the final performance of the trained model.

As already mentioned, each Approach described on **Section 3.3** has their own peculiarities. These can be different on: 1) Data Augmentation process; 2) moments to apply Data Augmentation; 3) pre- or post-processing instructions; 4) evaluation process; 5) metrics to evaluate. These differences are described on **Sub Section 4.4.9**. However, in all different approaches, the final model is the output of the k-fold Cross-Validation, where the final model is trained with all the data.

*Model Validation*

Although model's training and evaluation is a very important step in the process of building a Machine Learning model, validation of that same model is of great importance. There are several tests that can be performed to assess the model.

One simple evaluation to perform is to understand what is influencing model's choices and mislead possible cases where model is picking suspicious features or variables. For this, a simple technique is to analyse model's Features Importance, where for each feature a value is assigned which represents it's relative or absolute importance in the set of all features. Depending on the used algorithm to build the model, for example XGBoost [13], this implementation is straight forward, however, some algorithms does not support this task at all.

*Logs Persistence*

A very important practice is to keep track of the results of every single experience, where these can differentiate on multiple aspects or one simple configuration. Change model's hyperparameters, train and test split proportions or even random processes' seed, all these changes, no matter how simple or irrelevant they may seem, if a history of these experiences is maintained we can easily guide our work to not repeat and try new experiences or understand what leaded to the final results.

[13] Xgboost.readthedocs.io. (2019). XGBoost Documentation xgboost 1.0.0-SNAPSHOT documentation. [online] Available at: https://xgboost.readthedocs.io/en/latest/ [Accessed 30 Aug. 2019].

Having this in mind, all experiences were logged, resulting in a *Log File* in *text format*. Each *Log File* contains the following information:

- *Log* description;
- Path of input data for modelling;
- Machine Learning algorithm;
- Evaluation metric for Hyperparameter Optimization;
- Parameters settings for k-fold Cross-Validation search;
- Number of total possible combinations to search;
- Number o $k$ folds used on Hyperparameter Optimization;
- Random Search configuration;
- Mean CV score of Hyperparameter Tunning;
- Best found configuration of hyperparameters;
- Number of $k$ folds used on training model's CV;
- Data Augmentation configurations used;
- Dimensions of each fold during model training;
- Mean CV score of model training step;
- Evaluation score of the final model.

All the generated log files are attached in the end on **Listings Sub Section C.1**.

### 4.4.9 *Modelling approaches*

*Approach I*

In Approach I - Multiple Independent Supervised Trait Regressors described on **Section 3.3**, the implementation of Data Augmentation requires almost no additional work. This step of Data Augmentation is performed during each iteration of the k-fold Cross-Validation training. For this, after replicas are generated and the new scales are generated, the labels for those records are adjusted. In the case of *Approach I - Multiple Independent Supervised Trait Regressors (Approach I)*, the labels are each trait' evaluation score, for these, the scores themselves are updated according to [2]. After applying Data Augmentation all the pre-added auxiliary columns are removed so they are not considered during model training.

In order to evaluate the trained model, k-fold Cross-Validation is being used. For this effect, for each iteration of this method, a specific performance metric is calculated. After all iterations have finished, the performance scores mean is calculated which is then associated to the model as the final performance metric. For this Approach I, the proposed performance metrics are *Mean Absolute Error (MAE)* and *Root Mean Square Error (RMSE)*. However, as described on **Sub Section 4.3.5**, in this particular case, RMSE is the most suitable performance metric. So, RMSE was used also for model evaluation, being calculated during each iteration of the *k-fold Cross-Validation (CV)* model training process but it was also used as objective function [14] in the Machine Learning algorithm.

*Approach II*

Similar to what was performed for Approach I, in Approach II - Multiple Independent Supervised Scale Regressors, the implementation of Data Augmentation requires almost no additional work. During each iteration of the k-fold Cross-Validation training, *Data Augmentation (DA)* step is performed[15]. After both replicas and new scales are generated it is also required to update the labels of these new generated instances. In the case of *Approach II - Multiple Independent Supervised Scale Regressors (Approach II)*, the labels are each words' selected scales, which becomes much easier to update the new labels, since the labels are the new generated scales. After applying Data Augmentation all the pre-added auxiliary columns are removed so they are not considered during model training.

Since k-fold Cross-Validation is being used to evaluate the trained model, for each iteration of this method, a specific performance metric is calculated. After all iterations have finished, the performance scores mean is calculated which is then associated to the model as the final performance metric. However, this approach's model must be evaluated on two different situations: *a)* the quality of the words' scales predictions; *b)* the quality of the final traits' scores, after applying the method on [2].

Both MAE and RMSE are proposed. However, as described on **Sub Section 4.3.5**, in this particular case, RMSE is the most suitable performance metric. So, RMSE was used both for evaluation and as objective function [16] in the Machine Learning algorithm.

[14] Specifies the learning task and the corresponding learning objective

[15] However, in the first implementations of DA, this process was performed before k-fold Cross-Validation.

[16] Specifies the learning task and the corresponding learning objective

*Approach III*

Unlike previous described approaches, in *Approach III - Supervised Multi-class Bin Classification (Approach III)* the implementation of Data Augmentation requires extra transformation operations. Although, DA is still performed during each iteration of CV training[17].

Ending each iteration's DA step, it is required to update the labels of each new generated record. In the case of Approach III, this is performed in two consecutive steps: *a)* calculate each trait's score; *b)* recalculate binarization of these scores. The first step *a)* is performed according to [2], just like in Approach I. The second and final step *b)* is performed just as described in **Sub Section 4.4.1**. After applying Data Augmentation all the pre-added auxiliary columns are removed so they are not considered during model training.

The trained model is being evaluated during k-fold Cross-Validation, where a specific performance metric is calculated for each iteration's result. The associated performance metric to this model is the calculated mean of all iterations' scores.

Since Approach III is a Multi-class Classification problem, the previous used metrics cannot be used, for this, multiple evaluation metrics for multiple purposes and insides are being used.

As objective function, *Area Under Curve (AUC)* is being used, this is a proper metric because beyond being fully supported by the used framework it also prevents overfitting on high skewed populations of data, which is the case as described in **Sub Section 4.4.1**. However, to evaluate model's performance, two metrics are being used, ***F1 Score (F1)*** (calculates as **Equation 4**) and **Mean Error** (as **Equation 5**). The model's Confusion Matrix, also known as Error Matrix, is also calculated and analysed, however, it is not used to determine if a certain model is better or not than other.

$$F1 = 2 * \frac{prec * rec}{prec + rec} \tag{4a}$$

$$prec = \frac{TP}{TP + FP} \tag{4b}$$

$$rec = \frac{TP}{FN + TP} \tag{4c}$$

where:

$F1$ = F1 Score

$prec$ = Precision

$rec$ = Recall

$TP$ = Number of true positives

$FP$ = Number of false positives

[17] Although, like previous approaches, in the first implementations of DA, this process was performed before k-fold Cross-Validation.

$FN$ = Number of false negatives

$$ME = \frac{TE}{TR} \tag{5}$$

where:

$ME$ = Mean Error
$TE$ = Total number of wrongly classified records
$TR$ = Total number of records

# 5

# EXPERIMENTS AND RESULTS

## 5.1 EXPERIMENTAL SETUP

A very important aspect of every study and scientific work is to ensure that experiments can be replicated by third parties. For that, it is of great importance to describe, document and present all relevant aspects and details of the setup and configurations used to achieve the results described in the next sub sections.

### 5.1.1 *Computational Resources*

A very important aspect of every conducted experiment, is the available computational capabilities. In the following sections performance metrics are referenced, like models training duration, these are associated to a specific computational configuration. The following performance values may vary by configuration to configuration, so it is important to described the used one.

For this experiment, a **MacBook Pro 13-inch, 2019, Four Thunderbolt 3 ports** was used, equipped with a **2,8 GHz Intel Core i7 processor**, **16 GB 2133 MHz LPDDR3 memory**, a **graphic card Intel Iris Plus Graphics 655 1536 MB** and a **500GB SSD**.

### 5.1.2 *Data Split*

Although one of the greatest problems is the low amount of data, DA (described in the previous chapters) reduces the impact. In order to be able to mitigate possible situations of overfitting, even with low amounts of data, the original data set was splitted into two different subsets: 1) training and 2) test set.

One of the most important and significant steps, is *Hyperparameter Optimization (Hyperparameter Tunning)*, this step uses this subset of data to find the best hyperparameters for the model.

Test set is the subset of data which allows to mitigate possible cases of overfitting. Since this data is not used in any part of the modelling and so never seen by the model, it helps to understand how the final model behaves on unseen data. However, it is important to understand that the performance of the model on this data, must not be used to choose the best model, instead, it should be used the score of the model in the validation set (computed during k-fold Cross-Validation). If this happened, this would add bias creating a human error, which can also be called of overfitting.

The split of these subsets was performed with a proportion of 80-20%. These proportions were used because of the problem of having low amounts of data. The original dataset has a total of 243 records (after all filtering, data preparation and feature engineering), which become a total of 195 records for training and 48 for testing the final models.

### 5.1.3 *Hyperparameter Optimization*

As previously referred, Hyperparameter Optimization is one of the most important steps in the process of building a Machine Learning model. Different choices must be made, all with great importance: 1) the method to look for the best hyperparameters; 2) the objective function or evaluation metric used to converge to the best combination of hyperparameters; 3) the list of values to try and explore.

As previously described, Random Search was the method used to look for the best combination of hyperparameters. This was the method used for all approaches, independently of the Modelling or the Data Augmentation approach.

The objective function or evaluation metric used to converge to the best combination depends on the Modelling approach. For both Approach I - Multiple Independent Supervised Trait Regressors and Approach II - Multiple Independent Supervised Scale Regressors, regression with squared loss was used as objective function and Root Mean Square Error as evaluation metric. On the other hand, for Approach III - Supervised Multi-class Bin Classification Area Under Curve is used as evaluation metric.

Although Random Search searches for the best hyperparameters combination, it is required manual configuration. Even after configuring a list of hyperparameters to search for the best combination, it can require manual analysis and even an update of the list of hyperparameters to use for the search. So, after looking for the top combination, every single time an analysis was performed. In case the found configuration included any value of the extremes of values to any of the hyperparameters, this intervals of values were moved, either up or down, depending on what extreme was used. This process required multiple runs and multiple experiments. The final list of hyperparameters to perform Random Search was found and can be consulted on **Table 5** for both Approach I and Approach II. For the Approach III it is listed on **Table 6**. If some hyper parameter it is not indicated

| Hyper parameters | List of values used for Random Search |
|---|---|
| learning_rate | 0.001, 0.01 |
| max_depth | 10, 12, 14 |
| min_child_weight | 6, 7, 8 |
| gamma | 0.04, 0.05, 0.06 |
| colsample_bytree | 0.3, 0.4, 0.5 |
| eta | 0.05, 0.1, 0.15 |
| n_estimators | 500, 600, 700, 800 |
| eval_metric | 'rmse' |
| objective | 'reg:squarederror' |

Table 5.: List of hyper parameters and respective list of values used for Random Search for Modelling Approach I - Multiple Independent Supervised Trait Regressors and Approach II - Multiple Independent Supervised Scale Regressors.

on the tables, it means the default of the *eXtreme Gradient Boosting (XGBoost)* package was used.

Although the total number of possible combinations is very high, 1944, a total of 500 random combinations were executed. This process, used k-fold Cross-Validation to find the best combination with *k*=2

### 5.1.4 *k-fold Cross-Validation*

In all approaches, the total number of folds in the process of k-fold Cross-Validation was *k*=4. The value for k was chosen such that each train and test folds were large enough to be statistically representative of the broader dataset.

### 5.1.5 *Data Augmentation*

In each iteration of k-fold Cross-Validation, the following configurations of Data Augmentation were used.

On Data Augmentation I, every single record was used to create a total of 40 new records, where in each new record one and only one records changed.

On Data Augmentation II, for each record, a total of 40 new records were generated. However, in each new record a total of 3 words differentiate.

Finally, on Data Augmentation III, the most elaborated process of DA, for each record the probability of using a pre found pattern, on Pattern Mining tasks, is of 60%. Each instance

| Hyper parameters | List of values used for Random Search |
|---|---|
| learning_rate | 0.001, 0.01 |
| max_depth | 10, 12, 14 |
| min_child_weight | 6, 7, 8 |
| gamma | 0.04, 0.05, 0.06 |
| colsample_bytree | 0.3, 0.4, 0.5 |
| eta | 0.05, 0.1, 0.15 |
| n_estimators | 500, 600, 700, 800 |
| eval_metric | 'auc' |

Table 6.: List of hyper parameters and respective list of values used for Random Search for Modelling Approach III - Supervised Multi-class Bin Classification.

can generate a minimum of 200 new records and a maximum of 300 new records, where each record differentiates on a minimum of 6 words and a total of 12 words. The seed used for all these processes was *42*.

After DA, each model (of each approach on different DA iterations) was trained with a dataset with different dimensions. Approach I was trained with 728, 18.360, 12.300 and 75.800 records with no DA, DA I, DA II and DA III respectively. Approach II used 728, 18.360, 29.848 and 76.000 records in the same DA iterations. Lastly, Approach III used, 728, 18.360, 29.848 and 75.600 records for no DA both with 3 and 5 bins , DA I, DA II and DA III respectively.

## 5.2 RESULTS

Firstly, a general revision of the found patterns resulted on Pattern Mining are exposed. In the following sub-sections, the results achieved on each of the proposed modelling approaches and Data Augmentation iterations are presented and described.

### 5.2.1 *Approach I - Multiple Independent Supervised Trait Regressors*

Approach I - Multiple Independent Supervised Trait Regressors is the simpler proposed approach, however, it still has some complexity in terms o architecture where in fact 5 distinct models, one for each trait, are trained. These models have no interdependence.

In terms of training time, due to its simplicity, this is one of the fastest training process. The entire process takes different time to run for each DA iteration. **No DA**, since it is very straightforward, it is very fast, taking less than *10 minutes* to train. **DA I**, although with more data, it is still quite fast, taking around *14 minutes*. Since **DA II** and **DA III** requires significantly more processing, these take around *30* and *50 minutes* respectively. In **Figure 29** and **Figure 30** are presented the results for this approach with different Data Augmentation iterations. For a more accurate analysis of these results **Table 7** is presented. On each different Data Augmentation iteration there is a significant improvement relative to the previous one. However, this does not happens on **DA I**, which is caused by overfitting. Since the model is being trained with a lot of similar data, due to the poor DA method, and then tested against a lot of similar data too but completely different from the training data. As we can see, both MAE and  of **Folds' Mean** are much lower than the values of **Test Score**, which is an indicator of overfitting.

Apart from **DA I**, all other iterations present small differences between **Folds' Mean** and **Test Score**, which is very good. Oppositely to **DA I**, which presents high differences, these small differences model shows no indications of overfitting. With an impressive result, even with very good results, **DA III** with a RMSE of *5.91*, it only has a difference of *0.1* between **Folds' Mean** and **Test Score**. It is important to notice that the differences between these two subsets represents differences lower than *0.5* units of measure of each trait, which is a low variation.

Independently of DA iteration or **Folds' Mean** and **Test Score**, MAE is always inferior to RMSE. This is an indication that probably there are several outliers in the data, however, these still have to be accounted.

For this approach, from an initial RMSE score of *10.37* and final score of *5.91*, there was a significant improvement of *4.46* units of measure. On an average population, without considering outliers, this model has a score of *4.68*, which is a very good precision, repre-

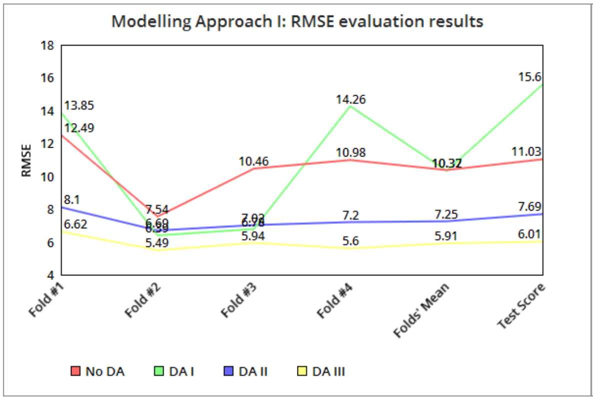
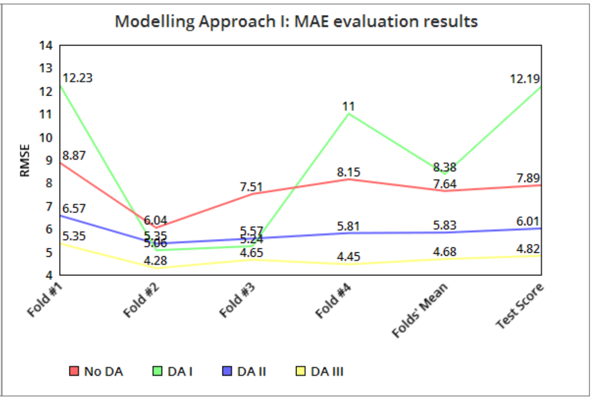


Figure 29.: RMSE evaluation of Approach trained models.



IFigure 30.: MAE evaluation of Approach I trained models.

senting an error around 7%. The final score of this approach to use and compare with other approaches it is an RMSE of *5.91*, which is an error around 9%.

In the end of this document, the results of feature importance for each of the reported models are listed on **Section B.2**, except for the best model which is below (**Figure 31**).

The first model, with **No DA**, has its features importance listed on **Table 42**. As can be seen, considering **Table 10**, the list of words with more importance for each trait does not completely match with the table values. Although this is not necessarily bad, this presents some concerns. One to notice, is the trait **conscientiousness**, which presents a completely different list of relevant words.

One of the most important differences on the other models, **DA II** and **DA III** reported on **Table 44** and **Table 31** respectively, is that there are much less relevant features for each trait but each one of them presents much higher importance. Another difference is that each of the most important features does match with the list of words on the referenced table.

| Fold # | Metric | Data Agumentation Approach | | | |
|---|---|---|---|---|---|
| | | No DA | DA I | DA II | DA III |
| 1 | MAE | 8.87 | 12.23 | 6.57 | 5.35 |
| | RMSE | 12.49 | 13.85 | 8.10 | 6.62 |
| 2 | MAE | 6.04 | 5.06 | 5.35 | 4.28 |
| | RMSE | 7.54 | 6.39 | 6.69 | 5.49 |
| 3 | MAE | 7.51 | 5.24 | 5.57 | 4.65 |
| | RMSE | 10.46 | 6.78 | 7.02 | 5.94 |
| 4 | MAE | 8.15 | 11.00 | 5.81 | 4.45 |
| | RMSE | 10.98 | 14.26 | 7.20 | 5.60 |
| Folds' Mean | MAE | 7.64 | 8.38 | 5.83 | 4.68 |
| | RMSE | 10.37 | 10.32 | 7.25 | 5.91 |
| Test Score | MAE | 7.89 | 12.19 | 6.01 | 4.82 |
| | RMSE | 11.03 | 15.60 | 7.69 | 6.01 |

Table 7.: Modelling Approach I: metrics evaluation results.

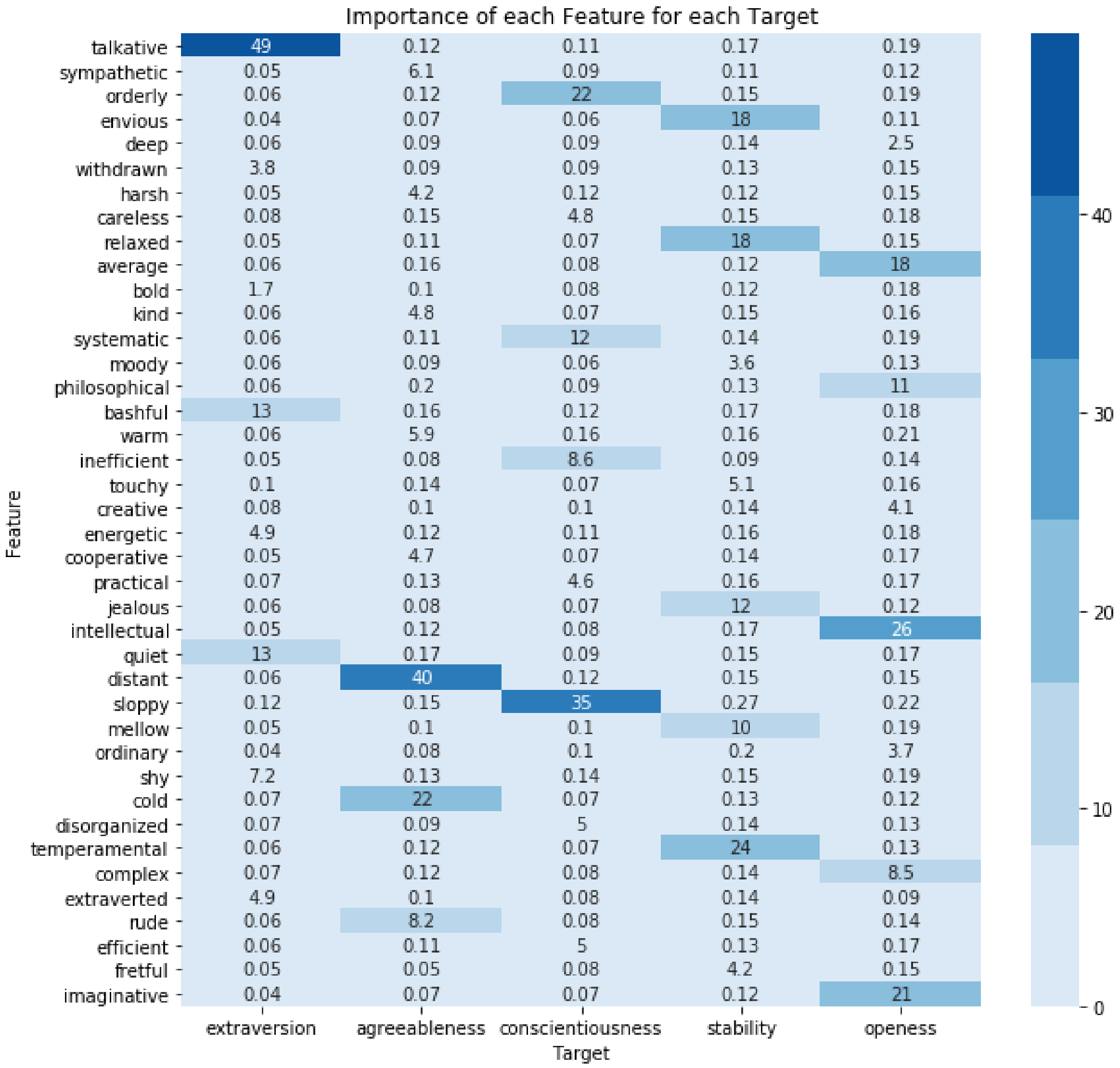


Figure 31.: Features importance on Approach I with Data Augmentation implementation III model.

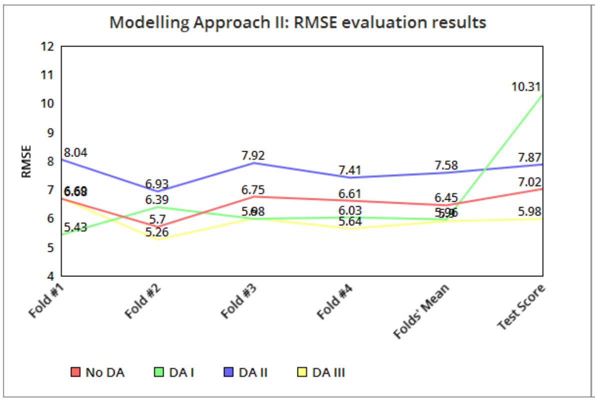


Figure 32.: RMSE evaluation of Approach II trained models.

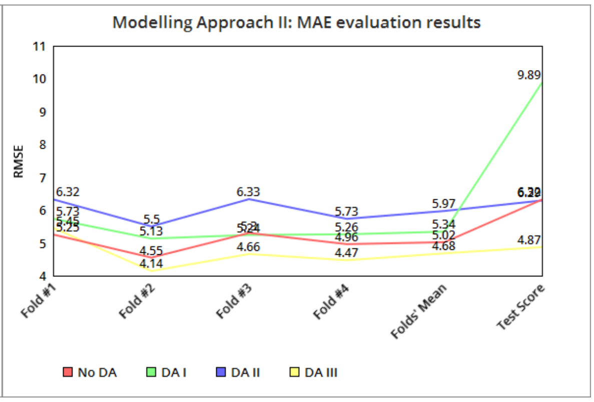


Figure 33.: MAE evaluation of Approach II trained models.

### 5.2.2 *Approach II - Multiple Independent Supervised Scale Regressors*

Approach II - Multiple Independent Supervised Scale Regressors it is one of the most computationally costly of all 3 approaches. This approach requires a total of 40 models to train. It is to notice that on each fold of k-fold Cross-Validation there are 40 models to train. Once more, these models have no interdependence.

In terms of training time, since it has an higher amount of models to train this become much slower than the previous described approach. It still requires different amounts of time to run for different DA iterations.

**No DA**, since it is very straightforward, it is very fast, taking less than *20 minutes* to train. With more data to use to train, **DA I** starts to take more time to train, around *35 minutes*. Once more, since requiring significantly more processing to generate new data on DA and the high amount of models to build, **DA II** and **DA III** require around *110* and *120 minutes* respectively.

In **Figure 32** and **Figure 33** are presented the results for this approach with different Data Augmentation iterations. For a more accurate analysis of these results **Table 8** is presented. It is to notice the columns **S** and **T**, where **S** is the evaluation of the output of the trained models (individual words' scales) and **T** is the evaluation of the traits' scores after converting models' outputs into traits' scores. These step is of high importance since it allows to compare Approach I and Approach II models performances.

On each different Data Augmentation iteration there is no significant improvement relative to the previous one. In fact, there is only an improvement from **No DA** to **DA III**.

On **DA I**, although it suffers a decrease of performance, it is not the worst model. However, since this is being trained with data with low variance and then tested against a completely different subset of data, it reveals high indices of overfitting. As we can see,

| Fold # | Metric | Data Agumentation Approach | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | No DA | | DA I | | DA II | | DA III | |
| | | S | T | S | T | S | T | S | T |
| 1 | MAE | 1.59 | 5.25 | 2.00 | 5.73 | 1.86 | 6.32 | 1.60 | 5.45 |
| | RMSE | 1.93 | 6.68 | 1.12 | 5.43 | 2.19 | 8.04 | 1.89 | 6.69 |
| 2 | MAE | 1.67 | 4.55 | 2.06 | 5.13 | 2.02 | 5.50 | 1.55 | 4.14 |
| | RMSE | 2.02 | 5.70 | 1.06 | 6.39 | 2.38 | 6.93 | 1.88 | 5.26 |
| 3 | MAE | 1.57 | 5.30 | 1.86 | 5.24 | 1.86 | 6.33 | 1.51 | 4.66 |
| | RMSE | 1.92 | 6.75 | 1.71 | 5.98 | 2.21 | 7.92 | 1.81 | 6.00 |
| 4 | MAE | 1.51 | 4.96 | 1.93 | 5.26 | 1.73 | 5.73 | 1.48 | 4.47 |
| | RMSE | 1.91 | 6.61 | 1.32 | 6.03 | 2.07 | 7.41 | 1.76 | 5.64 |
| Folds' Mean | MAE | 1.59 | 5.02 | 1.96 | 5.34 | 1.87 | 5.97 | 1.54 | 4.68 |
| | RMSE | 1.95 | 6.45 | 1.96 | 5.96 | 2.21 | 7.58 | 1.84 | 5.90 |
| Test Score | MAE | 1.68 | 6.32 | 2.32 | 9.89 | 1.94 | 6.29 | 1.61 | 4.87 |
| | RMSE | 2.58 | 7.02 | 2.71 | 10.31 | 2.46 | 7.87 | 1.92 | 5.98 |

Table 8.: Modelling Approach II: metrics evaluation results. Columns: **S**: Scores on predicting words' scales; **T**: Scores after converting words' scales to traits' scores.

both MAE and RMSE of **Folds' Mean** are much lower than the values of **Test Score**, which is an indicator of this phenomenon.

**DA II**, although a supposedly better DA method, it also presents a worse score comparing to **No DA**. However, even with a worst performance than **DA I**, this DA method presents similar performance scores both on **Folds' Mean** and **Test Score**, which indicates no overfitting of the model.

Finally, **DA III** although it does not present great performance improvements compared to **No DA**, it has several improvements on **Test Score**. Although previously stated this should not be used to compare models, these lower differences shows this is a better model on overfitting, since it scores approximately equally on both subsets of data. It is important to notice that, the differences between these two subsets represents differences lower than *0.1* units of measure (RMSE) of each traits, which is a very low variation.

Just like the previous approach, MAE is lower than RMSE, being this an indication of the existing of several outliers in the data.

For this approach, from an initial RMSE score of *6.45* and final score of *5.90*, there was no significant improvement, with an improvement of *0.55*. Considering a population without outliers, this model has a score of *4.47*, which is a very good precision, representing an error below 7%. The final score of this approach to use and compare with other approaches it is an RMSE of *5.90*, which is an error around 9%.

Approach II's features importance analysis results of the described models are also reported on **Section B.2**, except for the best model which is presented below. Unlike the previous described Approach I, it is not expected to have multiple features with high relevance per target. In fact, since the target is the word's scale, it is expected the feature with highest relevance to be the word's selection itself.

**Table 45** lists the analysis of **No DA**. Even on this first basic model this behaviour is strongly visible, where the word's selection has much greater importance than any other feature. However, other feature still have importance. The evolution from the first model to **DA I** and **DA II** models can be easily seen, on **Table 46** and **Table 47** respectively, where we have even higher importance on the respective word's selection and less in the remaining words. In the end, the final model presents very high importance on the respective word and residual importance in the remaining ones.

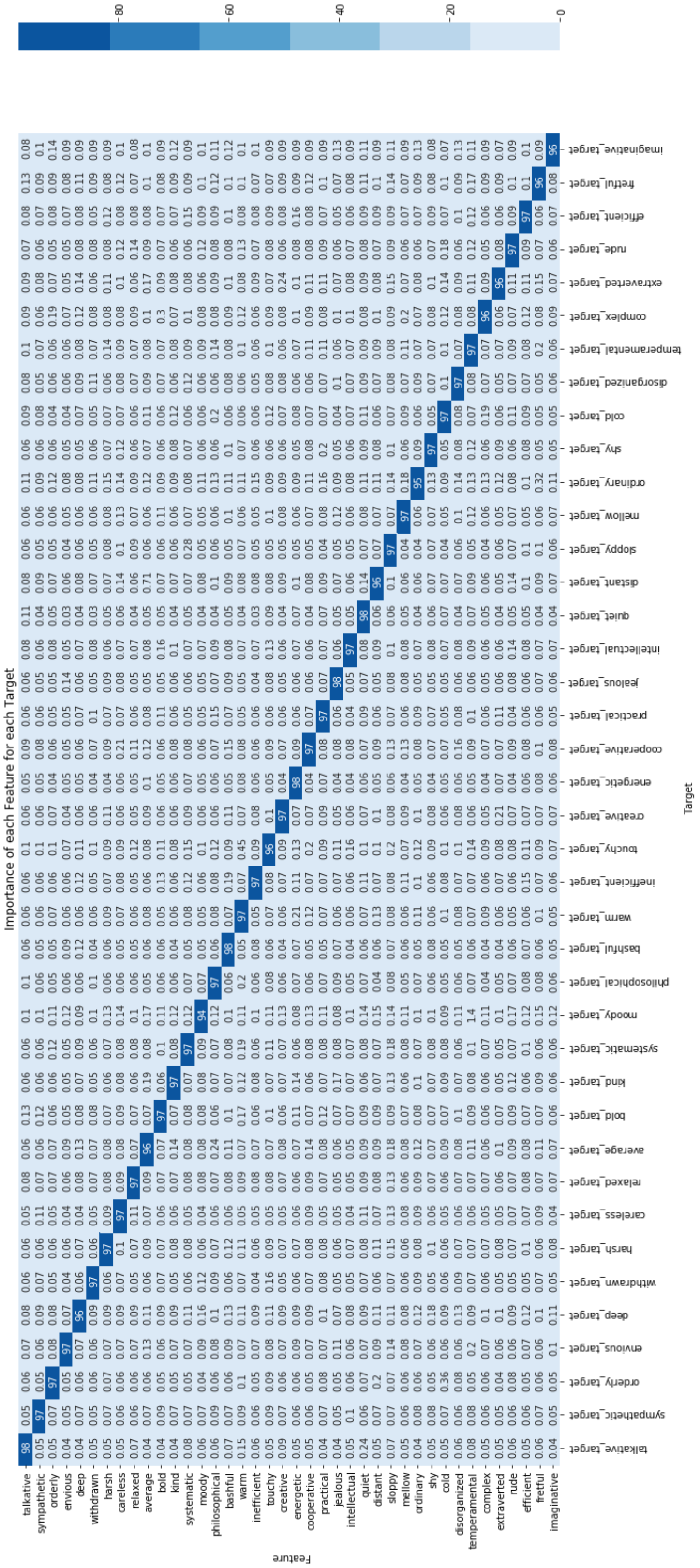


Figure 34.: Features importance on Approach II with Data Augmentation implementation III model.

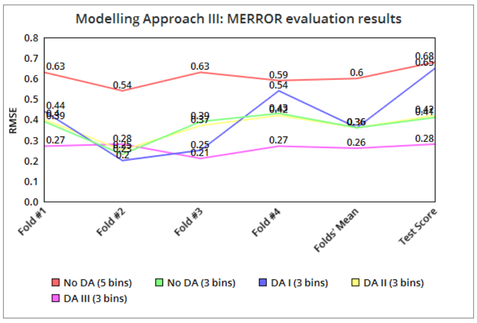


Figure 35.: MERROR evaluation of Approach III trained models.

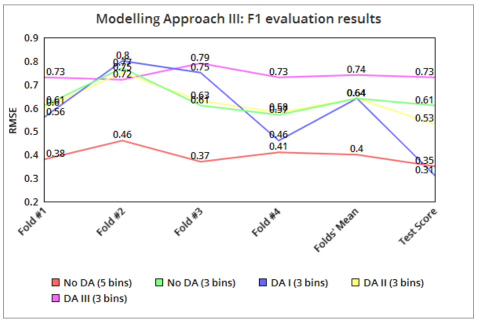


Figure 36.: F1 evaluation of Approach III trained models

### 5.2.3 *Approach III - Supervised Multi-class Bin Classification*

Approach III - Supervised Multi-class Bin Classification it is the only Classification approach. Like Approach I, this requires a total of 5 distinct models to train, one for each trait. To notice these models have no interdependence.

In terms of training time, due to its simplicity, this is a fast training process. The entire process takes different time to run for each DA iteration.

**No DA**, since it is very straightforward, it is very fast, taking around *20 minutes* to train. **DA I** taking around *35 minutes*, **DA II** and **DA III** requiring more processing, take around *50* and *80 minutes* respectively.

**Figure 35** and **Figure 36** present the results for this approach with different Data Augmentation iterations. For a more accurate analysis of these results **Table 9** is presented. Also, there are represented experiences both with *5* and *3 bins*. One experience with *5 bins*, **No DA**, and 4 different experiences with *3 bins*, similarly to previous approaches, **No DA**, **DA I**, **DA II** and **DA III**.

Although multiple experiences, no improvements are acquired from iteration to iteration until **DA III**. One important aspect to notice, as previously stated, there are a great improvement by using *3 bins* instead of *5 bins*, as can be seen by the difference of the performance scores on experiences **No DA 5 bins** and **No DA 3 bins**. Not only it shows better performance on **Folds' Mean** but it also has smaller differences between these values and **Test Score**. This was one of the main causes to proceed with a total of *3 bins*.

Contrary to previous approaches, **DA I**, although still showing signs of overfitting, it remains with similar performance scores. However, since it continues to use the same DA methods, this is trained on poor data that does not represents the all population/reality of data. This is one of the greatest differences, where both MAE and RMSE of **Test Score** are around the double of **Folds' Mean**, showing very high indications of overfitting.

| Fold # | Metric | Data Agumentation Approach | | | | |
|---|---|---|---|---|---|---|
| | | No DA<br>5 bins | No DA<br>3 bins | DA I<br>3 bins | DA II<br>3 bins | DA III<br>3 bins |
| 1 | F1 | 0.38 | 0.61 | 0.56 | 0.60 | 0.73 |
| | MERROR | 0.63 | 0.39 | 0.44 | 0.40 | 0.27 |
| 2 | F1 | 0.46 | 0.77 | 0.80 | 0.75 | 0.72 |
| | MERROR | 0.54 | 0.23 | 0.20 | 0.25 | 0.28 |
| 3 | F1 | 0.37 | 0.61 | 0.75 | 0.63 | 0.79 |
| | MERROR | 0.63 | 0.39 | 0.25 | 0.37 | 0.21 |
| 4 | F1 | 0.41 | 0.57 | 0.46 | 0.58 | 0.73 |
| | MERROR | 0.59 | 0.43 | 0.54 | 0.42 | 0.27 |
| Folds' Mean | F1 | 0.40 | 0.64 | 0.64 | 0.64 | 0.74 |
| | MERROR | 0.60 | 0.36 | 0.36 | 0.36 | 0.26 |
| Test Score | F1 | 0.35 | 0.61 | 0.31 | 0.53 | 0.73 |
| | MERROR | 0.68 | 0.41 | 0.65 | 0.42 | 0.28 |

Table 9.: Modelling Approach III: metrics evaluation results.

Although **DA II** shows no improvements when compared to **No DA**, it presents a much better results in the differences between **Folds' Mean** and **Test Score** comparing to **DA I**, showing that overfitting is successfully reduced.

With very good results, **DA III** presents a relevant improvement compared to the inital experiments. With a *MERROR* of *0.26*, meaning this model is correct on *74%* of predictions. This presents a difference between **Folds' Mean**m and **Test Score** of just *0.2* and *0.1* respectively, showing no signs of overfitting.

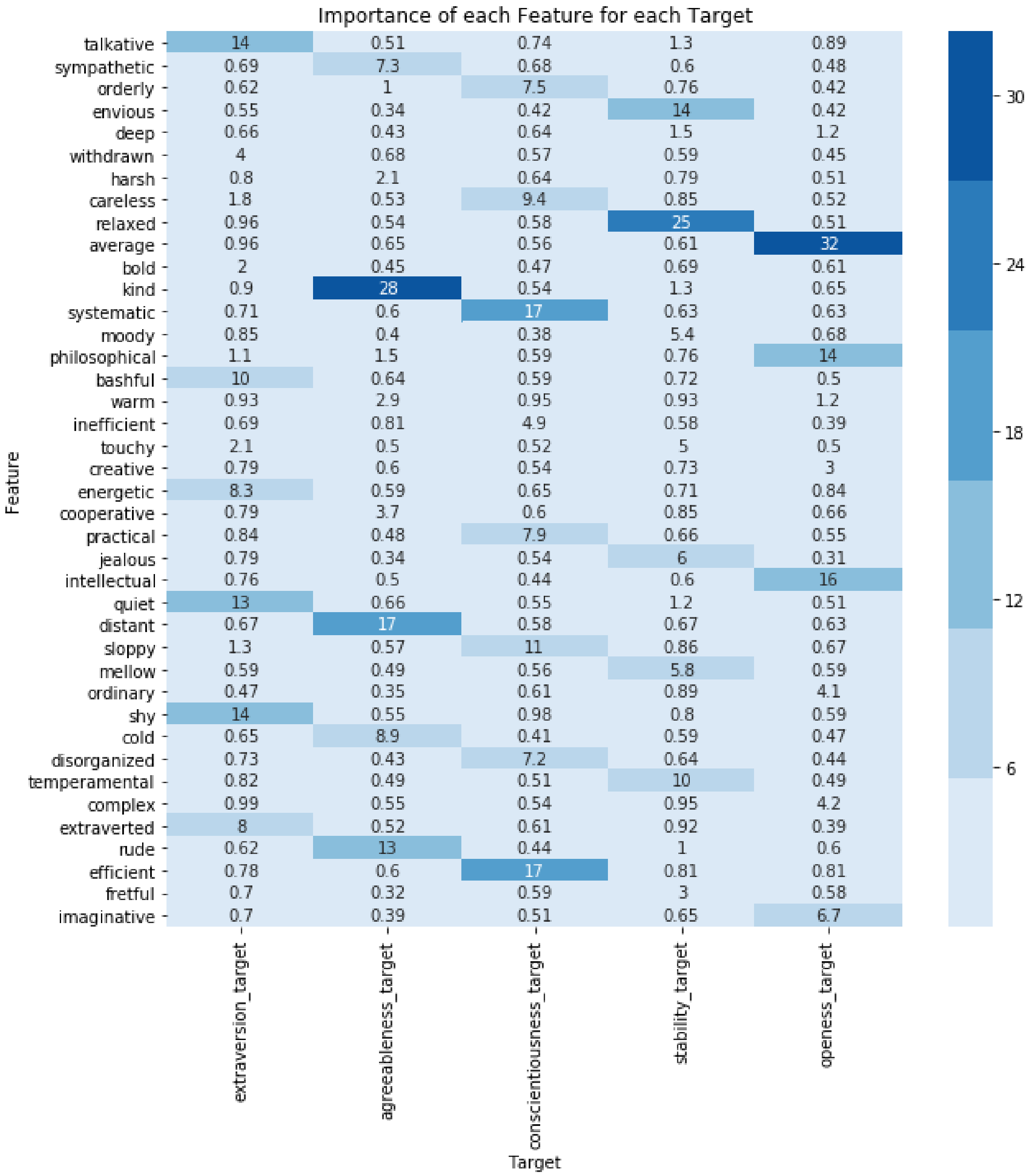
For this approach, from an initial F1 Score score of *0.40* and final score of *0.74*, there was a significant improvement of *0.34*.

Another metric to compare and simpler to translate into concrete results, it is the initial *MERROR* of *0.60* and final score of *0.26*, with an improvement of *0.34*.

The last approach's results, Approach III, are also reported in the end of this document on **Section B.2**, except for the best model.

**Table 48** and **Table 49** contain the results for the models with **No DA with 5 bins** and **No DA with 3 bins** respectively. These two models presents very distinct properties. The first one, presenting several words mismatching when compared with **Table 10** and drastically on *Extraversion* and *Agreeableness* traits. Besides that, there are multiple relevant features for each trait with no particular features standing out. On the other hand, although with no perfect behaviour, the second model, using *3 bins*, shows a few features standing out with less words mismatching. However, this still presents string mismatching on *Agreeableness* trait.

The first model with DA, **DA I with 3 bins**, shows a worst behaviour, showing less features standing out, more mismatching and no trait is specially well explained or characterised by any feature as can be seen on **Table 50**. **Table 51**, representing **DA II**, there are less more mismatching and a few words standing out, traits are not particularly explained by no specific groups of words. This does not happens in the final mode,  on **Table 37**. This model presents multiple and better standing out with low words mismatching. Even on the sub group of words which stand out some of them stand out, which shows a strong relation with each trait.



| Feature | extraversion_target | agreeableness_target | conscientiousness_target | stability_target | openess_target |
|---|---|---|---|---|---|
| talkative | 14 | 0.51 | 0.74 | 1.3 | 0.89 |
| sympathetic | 0.69 | 7.3 | 0.68 | 0.6 | 0.48 |
| orderly | 0.62 | 1 | 7.5 | 0.76 | 0.42 |
| envious | 0.55 | 0.34 | 0.42 | 14 | 0.42 |
| deep | 0.66 | 0.43 | 0.64 | 1.5 | 1.2 |
| withdrawn | 4 | 0.68 | 0.57 | 0.59 | 0.45 |
| harsh | 0.8 | 2.1 | 0.64 | 0.79 | 0.51 |
| careless | 1.8 | 0.53 | 9.4 | 0.85 | 0.52 |
| relaxed | 0.96 | 0.54 | 0.58 | 25 | 0.51 |
| average | 0.96 | 0.65 | 0.56 | 0.61 | 32 |
| bold | 2 | 0.45 | 0.47 | 0.69 | 0.61 |
| kind | 0.9 | 28 | 0.54 | 1.3 | 0.65 |
| systematic | 0.71 | 0.6 | 17 | 0.63 | 0.63 |
| moody | 0.85 | 0.4 | 0.38 | 5.4 | 0.68 |
| philosophical | 1.1 | 1.5 | 0.59 | 0.76 | 14 |
| bashful | 10 | 0.64 | 0.59 | 0.72 | 0.5 |
| warm | 0.93 | 2.9 | 0.95 | 0.93 | 1.2 |
| inefficient | 0.69 | 0.81 | 4.9 | 0.58 | 0.39 |
| touchy | 2.1 | 0.5 | 0.52 | 5 | 0.5 |
| creative | 0.79 | 0.6 | 0.54 | 0.73 | 3 |
| energetic | 8.3 | 0.59 | 0.65 | 0.71 | 0.84 |
| cooperative | 0.79 | 3.7 | 0.6 | 0.85 | 0.66 |
| practical | 0.84 | 0.48 | 7.9 | 0.66 | 0.55 |
| jealous | 0.79 | 0.34 | 0.54 | 6 | 0.31 |
| intellectual | 0.76 | 0.5 | 0.44 | 0.6 | 16 |
| quiet | 13 | 0.66 | 0.55 | 1.2 | 0.51 |
| distant | 0.67 | 17 | 0.58 | 0.67 | 0.63 |
| sloppy | 1.3 | 0.57 | 11 | 0.86 | 0.67 |
| mellow | 0.59 | 0.49 | 0.56 | 5.8 | 0.59 |
| ordinary | 0.47 | 0.35 | 0.61 | 0.89 | 4.1 |
| shy | 14 | 0.55 | 0.98 | 0.8 | 0.59 |
| cold | 0.65 | 8.9 | 0.41 | 0.59 | 0.47 |
| disorganized | 0.73 | 0.43 | 7.2 | 0.64 | 0.44 |
| temperamental | 0.82 | 0.49 | 0.51 | 10 | 0.49 |
| complex | 0.99 | 0.55 | 0.54 | 0.95 | 4.2 |
| extraverted | 8 | 0.52 | 0.61 | 0.92 | 0.39 |
| rude | 0.62 | 13 | 0.44 | 1 | 0.6 |
| efficient | 0.78 | 0.6 | 17 | 0.81 | 0.81 |
| fretful | 0.7 | 0.32 | 0.59 | 3 | 0.58 |
| imaginative | 0.7 | 0.39 | 0.51 | 0.65 | 6.7 |

Figure 37.: Features importance on Approach III with Data Augmentation implementation III model with 3 bins.

### 5.2.4 *Phased Partial Test*

Although this work does not contemplates the development of the previously described Phased Partial Test, in this subsection it is briefly described what relevant information is created in this work which could lead to a successful implementation.

As previously described, this *Phased Partial Test (PP Test)* would be able to reduce the amount of words the user is exposed. This would be able to do by: *a)* reduce the total amount of words; *b)* show the words with a certain order which would change according to the subject's word selection.

The initial reduction of words *a)* could be accomplished by using the result of the analysis of the most important features of each model. Approaches' final models includes a minimal subgroup of words with relevant importance for each target having the remaining ones some residual or minor importance. What is suggested is to use only those more important words. With this, it would be able to perform a significant reduction of the maximum number of words a subject would be exposed.

However, the exposure of words by groups of words *b)* requires much more complexity. For this, the found patterns during the pattern mining should be used. In this, each group of words exposed to the subject would in fact be related and created, grouped, by using these patterns. A group would be in fact a subgroup of multiple patterns. A selection of specific words would allow to automatically select other words not present in the exposed words but present in the related patters.

It is important to notice with this Phased Partial Test, a lower precision model would be created. However, it is not expected to have that significant reduction, since we would be using the most relevant features and found patterns. Although not developed, this could be a very interesting experiment to conduct in future work.

## 5.3 DISCUSSION

Several models and approaches were tested and experimented. Each one of them has its particularities in the development process. In order to understand which is the best model, it is important to keep in mind the advantages and disadvantages of each approach.

Approach I - Multiple Independent Supervised Trait Regressors and Approach II - Multiple Independent Supervised Scale Regressors present similar results in the evaluation of the respective best candidate models. However, it is important to notice the behaviour of each one the approaches.

Approach I revealed a constant improvement from iteration to iteration, every time the Data Augmentation method was improved, the performance of the model also improved significantly, indicating room for improvements. Another characteristic is how long this takes generate the new data and train the model, taking between *30* and *50 minutes*.

On the other hand, Approach II takes until the final iteration of DA to show any signs of improvement. Every other version of DA created worst models, and the one with improvements presented very small improvements, a total of *0.34* units of measurement. Another downside is how long it takes to run the entire process, taking between *110* and *120 minutes*.

When comparing Approach I and Approach II, the differences of each best candidate models' scores is not relevant, *0.01*, producing both the same results. However, with one of them showing signs of significant improvements and low amounts of time to run, the most successfully approach is Approach I - Multiple Independent Supervised Trait Regressors.

Approach III shows great results also. However, in a real scenario it does not allows to make a deep Psychological Assessment, it allows to get a general characterisation. It could be helpful in cases where the general characterisation of each person's trait is the most important to evaluate rather than the specific score of each trait.

In the end, Approach I - Multiple Independent Supervised Trait Regressors and Approach II - Multiple Independent Supervised Scale Regressors produced similar results with a Mean Absolute Error of *4.68* and a Root Mean Square Error around *5.90*. However, the first one presents some advantages relativity to the second one, which makes it the most successfully and promising approach, taking around *40* to *50 minutes* less to train. This presents more evidences of improvements.

Another approach, Approach III - Supervised Multi-class Bin Classification was also conceived, this produced a model with a **MERROR** of around *26%*, being this a great result.

Once more, it is never too much to mention the importance of the conducted analyses and developments of pattern mining on the dataset of words' selection. This enabled to build a more robust and significant Data Augmentation method allowing us to successfully build these solutions. With these, multiple solutions to the original problem were successively built, tested and evaluated.

Although not developed, a possible starting point to develop a Phased Partial Test is described. This method could be developed, by performing a reduction of exposed words using the result of features importance analysis. The grouping of words could be created using the found patterns of the pattern mining analysis. A minor precision reduction would be expected but not a huge reduction. This could possible guide the assessment into a wrong direction, by presenting specific words in specific order and automatically select specific words. This is one of the greatest challenges of a future development of this procedure.

# 6

# CONCLUSIONS AND FUTURE WORK

The conducted experiments during this entire work had multiple iterations, multiple periods of development and analysis. In the end, these produced very good results with high applicability, whether for future investigations or as a useful framework.

The initial proposed reduction of the The Big Five Factors The Mini-Marker Test, from a scales selection basis to a words' selection basis, was successfully completed. In fact, two different solutions with very high precision are provided and more solution capable of determining general traits' characteristics of personality.

In the end, it was proved, with some minor error associated, that it is possible to create an even greater reduction of the original Goldberg's Big Five Personality Test and the reduction The Mini-Marker Test, being introduced by this work a new successful reduction.

Although good results were accomplished, in the end of this work, more work exists to be done. This can be performed on different directions, improving the already built models or even investigate, design and create alternative approaches. In this work, a suggested Phased Partial Test is suggested, which can be one of the next steps.

The improvement of the presented models could be accomplished in multiple forms. One of the most important, should be to gather a lot more data to train the models. Different Machine Learning algorithms can be tested to solve this problem, which could lead to better results.

On the other hand, the suggested Phased Partial Test, although not aiming a performance increase, it would be able to create an even reduced version of this new presented Word Selection Personality Test. The starting point to develop this framework is described, giving not only the instructions but also the needed data to successfully accomplish it.

On a completely different side of improvements, source code improvements would be highly accepted. These improvements are mainly in the following aspects:

- Optimise the code for faster Data Augmentation;
- Generalise all the approaches to easily experiment multiple Machine Learning algorithms;

- Create a version of this source code easy to consult and integrate so it could be used on future work and experiences of different people, not only IT specialist but also of other areas of expertise.

This would be of great importance since it would allow to run more experiments in less time, allowing us to run even more experiments in the same period of time. It would also ensure that future people or even the authors themselves to easily contribute, consult and analyse the described developed work.

# BIBLIOGRAPHY

[1] Goldberg, L. (1992). The development of markers for the Big-Five factor structure. Psychological Assessment, 4(1), pp.26-42.

[2] Gerard Saucier (1994): Mini-Markers: A Brief Version of Goldberg's Unipolar Big-Five Markers, Journal of Personality Assessment, 63:3, 506-516.

[3] Hudek-Kneevi J, Kardum I. Five 2009. Five-factor personality dimensions and 3 health related personality constructs as predictors of health. Croat Med J. 2009;50(4):394402.

[4] Vianello, M., Schnabel, K., Sriram, N. and Nosek, B. (2013). Gender differences in implicit and explicit personality traits. Personality and Individual Differences, 55(8), pp.994-999.

[5] Ashton, M. and Lee, K. (2009). The HEXACO-60: A Short Measure of the Major Dimensions of Personality. Journal of Personality Assessment, 91(4), pp.340-345.

[6] Kelley, K. (2005). Using the Myers-Briggs Type Indicator. PsycCRITIQUES, 50(33).

[7] Kim, D. (2011). To analyze character's personality by using Enneagram. Cartoon and Animation Studies, s23(1), pp.35-50.

[8] Barlett, C. and Anderson, C. (2012). Direct and indirect relations between the Big 5 personality traits and aggressive and violent behavior. Personality and Individual Differences, 52(8), pp.870-875.

[9] Chapman, P. (2000). CRISP-DM 1.0. Step-by-step data mining guide: SPSS.

[10] Docs.microsoft.com. (2018). Team Data Science Process Documentation. [online] Available at: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/ [Accessed 23 Dec. 2018].

[11] Kotov, R., Gamez, W., Schmidt, F. and Watson, D. (2010). Linking big personality traits to anxiety, depressive, and substance use disorders: A meta-analysis. Psychological Bulletin, 136(5), pp.768-821.

[12] Zillig, L., Hemenover, S. and Dienstbier, R. (2002). What Do We Assess when We Assess a Big 5 Trait? A Content Analysis of the Affective, Behavioral, and Cognitive Processes Represented in Big 5 Personality Inventories. Personality and Social Psychology Bulletin, 28(6), pp.847-858.

[13] Judge, T., Higgins, C., Thoresen, C. and Barrick, M. (1999). THE BIG FIVE PERSONALITY TRAITS, GENERAL MENTAL ABILITY, AND CAREER SUCCESS ACROSS THE LIFE SPAN. Personnel Psychology, 52(3), pp.621-652.

[14] MOUNT, M., BARRICK, M., SCULLEN, S. and ROUNDS, J. (2005). HIGHER-ORDER DIMENSIONS OF THE BIG FIVE PERSONALITY TRAITS AND THE BIG SIX VOCATIONAL INTEREST TYPES. Personnel Psychology, 58(2), pp.447-478.

[15] McCrae, R. and John, O. (1992). An Introduction to the Five-Factor Model and Its Applications. Journal of Personality, 60(2), pp.175-215.

[16] Neal, A., Yeo, G., Koy, A. and Xiao, T. (2011). Predicting the form and direction of work role performance from the Big 5 model of personality traits. Journal of Organizational Behavior, 33(2), pp.175-192.

[17] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, . (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, pp.2825-2830.

[18] Caruana R. and Niculescu-Mizil, A.: An Empirical Comparison of Supervised Learning Algorithms. Proceedings of the 23rd Intl Conference on Machine Learning (2006)

[19] Bauer, E., Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36 (1/2), 105139.

[20] Dietterich T.G. (2000) Ensemble Methods in Machine Learning. In: Multiple Classifier Systems. MCS 2000. Lecture Notes in Computer Science, vol 1857. Springer, Berlin, Heidelberg

[21] Chee, CH., Jaafar, J., Aziz, I.A. et al. Artif Intell Rev (2018). https://doi.org/10.1007/s10462-018-9629-z

[22] Bergstra, James, and Yoshua Bengio. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, vol. 13, Feb. 2012, pp. 281305.

[23] Bergstra, James, et al. Algorithms for Hyper-Parameter Optimization. 2012.

[24] Snoek, Jasper, et al. Practical Bayesian Optimization of Machine Learning Algorithms. 2012.

[25] Chen, T., Guestrin, C., 2016. XGBoost: A scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Association for Computing Machinery, pp. 785794. doi:10.1145/2939672.2939785

[26] Yu, W., Na, Z., Fengxia, Y. and Yanping, G. (2018). Magnetic resonance imaging study of gray matter in schizophrenia based on XGBoost. Journal of Integrative Neuroscience, 17(4).

[27] Sahoo, D. and Balabantaray, R. (2019). Single-Sentence Compression using XGBoost. International Journal of Information Retrieval Research, 9(3), pp.1-11.

[28] Pesantez-Narvaez, J., Guillen, M. and Alcaiz, M. (2019). Predicting Motor Insurance Claims Using Telematics DataXGBoost versus Logistic Regression. Risks, 7(2), p.70.

[29] Ke, Guolin, et al. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. edited by I. Guyon et al., Curran Associates, Inc., 2017, pp. 3146–3154.

[30] Varma, S. Simon, R. (2006, February). Bias in error estimation when using cross-validation for model selection. BMC Bioinformatics, 7(1), 91.

[31] Santos, M., Soares, J., Abreu, P., Araujo, H. and Santos, J. (2018). Cross-Validation for Imbalanced Datasets: Avoiding Overoptimistic and Overfitting Approaches [Research Frontier]. IEEE Computational Intelligence Magazine, 13(4), pp.59-76.

[32] Scott M Lundberg, Gabriel G Erion, and Su-In Lee. Consistent individualized feature attribution for tree ensembles. arXiv preprint arXiv:1802.03888, 2018.

[33] Wang, Y. (2014). Study on Data Mining Techniques and Algorithms of Association Rules Data Mining. Applied Mechanics and Materials, 543-547, pp.2040-2044.

[34] Agrawal, R.; Mannila, H.; Srikant, R.; Toivonen, H.; and Verkamo, A. I. 1996. Fast Discovery of Association Rules. In Advances in Knowledge Discovery and Data Mining, AAAI Press, 307-328.

[35] Agrawal, R., and Srikant, R. 1994. Fast Algorithms for Min- ing Association Rules. IBM Research Report RJ9839, June 1994, IBM Almaden Research Center, San Jose, CA.

[36] van Dyk, D. and Meng, X. (2001). The Art of Data Augmentation. Journal of Computational and Graphical Statistics, 10(1), pp.1-50.

[37] Xie, Q., Dai, Z., ... Le, Q.V., 2019. Unsupervised Data Augmentation, in: NIPS Submission. pp. 115.

[38] Sun, Y., Wong, A.K.C., Kamel, M.S., 2009. Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23, 687719. doi:10.1142/S0218001409007326

[39] Olive, D.J., 2017. Linear regression, Linear Regression. Springer International Publishing. doi:10.1007/978-3-319-55252-1

[40] Heimann, P., Isaacs, S., 2018. Regression, in: Developments in Psychoanalysis. Taylor and Francis, pp. 169197. doi:10.4324/9780429473661

[41] Grégoire, G., 2015. Logistic regression, in: EAS Publications Series. EDP Sciences, pp. 89120. doi:10.1051/eas/1466008

[42] Zhou, Z.H., 2012. Ensemble methods: Foundations and algorithms, Ensemble Methods: Foundations and Algorithms. CRC Press. doi:10.1201/b12207

[43] Kozak, J., 2019. Ensemble methods, in: Studies in Computational Intelligence. Springer Verlag, pp. 107118. doi:10.1007/978-3-319-93752-6$_6$

[44] Dietterich, T.G., 2000. Ensemble methods in machine learning, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). pp. 115.

[45] Toivonen, H., 2017. Apriori Algorithm, in: Encyclopedia of Machine Learning and Data Mining. Springer US, pp. 6060. doi:10.1007/978-1-4899-7687-1$_2$7

# A

# SUPPORT MATERIAL

### A.0.1 *Words influence on each trait*

| Trait | Negative influence | Positive influence |
|---|---|---|
| **Extraversion** | shy quiet withdrawn bashful | extraverted talkative energetic bold |
| **Agreeableness** | cold harsh rude distant | kind cooperative sympathetic warm |
| **Conscientiousness** | disorganized careless inefficient sloppy | systematic practical efficient orderly |
| **Stability** | moody temperamental envious fretful jealous touchy | relaxed mellow |
| **Openness** | average ordinary | intellectual creative complex imaginative philosophical deep |

Table 10.: List of words with Negative or Positive influence on each Trait.

# B

# DETAILS OF RESULTS

## B.1 SELECTION OF WORDS' SCALES BY WORD SELECTION

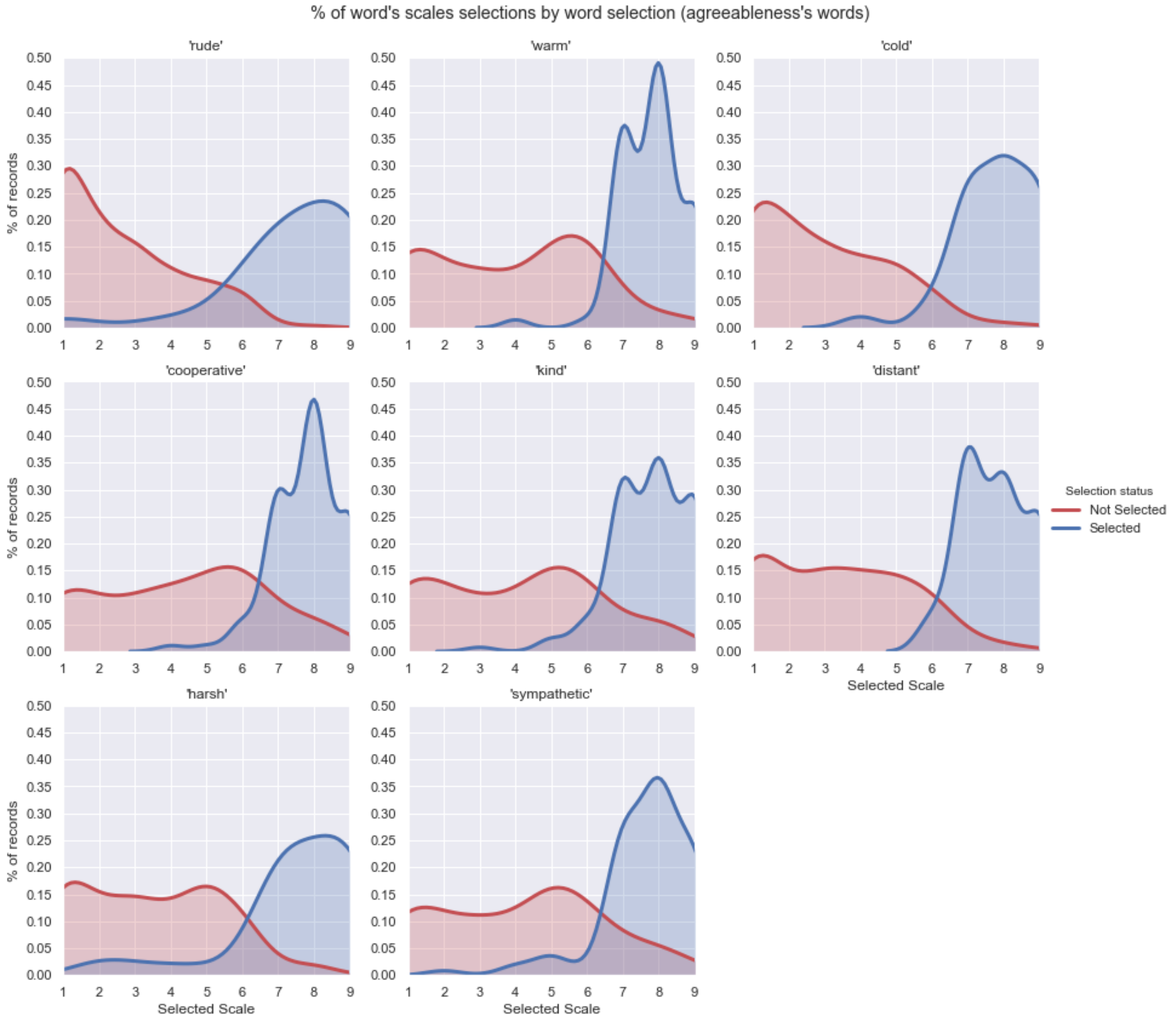


Figure 38.: Agreeableness % of word's scales selections by word selection

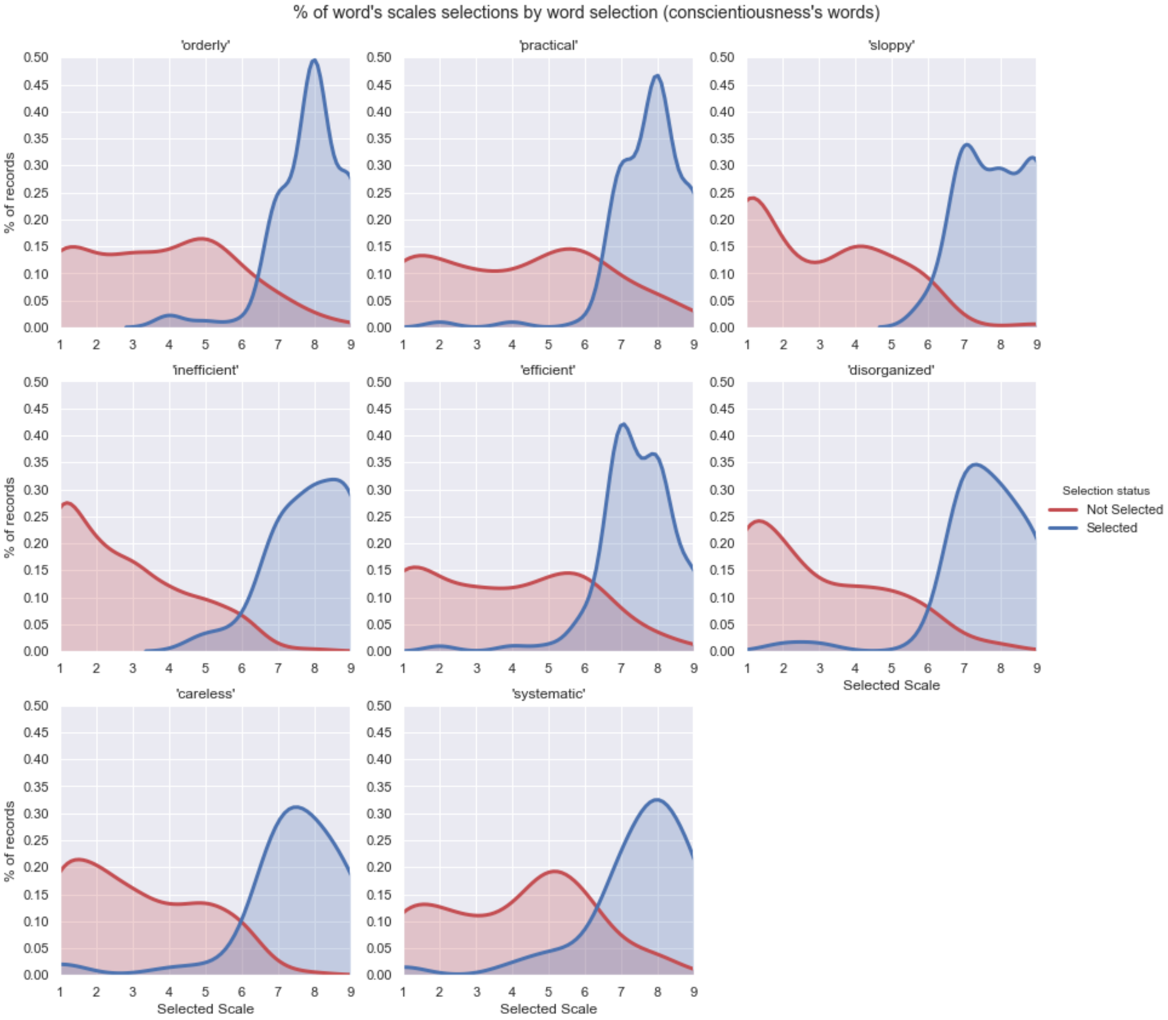


Figure 39.: Conscientiousness % of word's scales selections by word selection

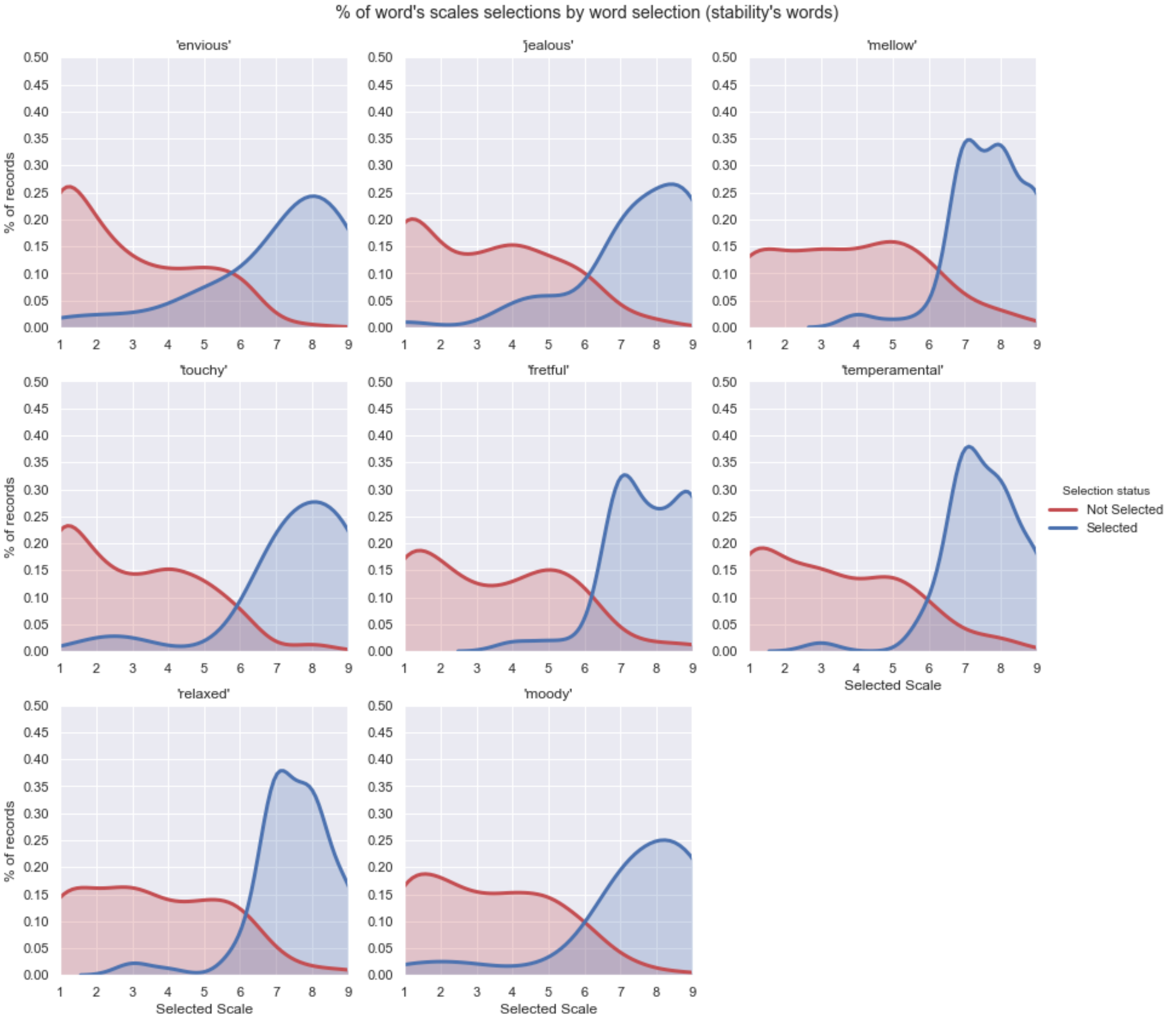


Figure 40.: Stability % of word's scales selections by word selection

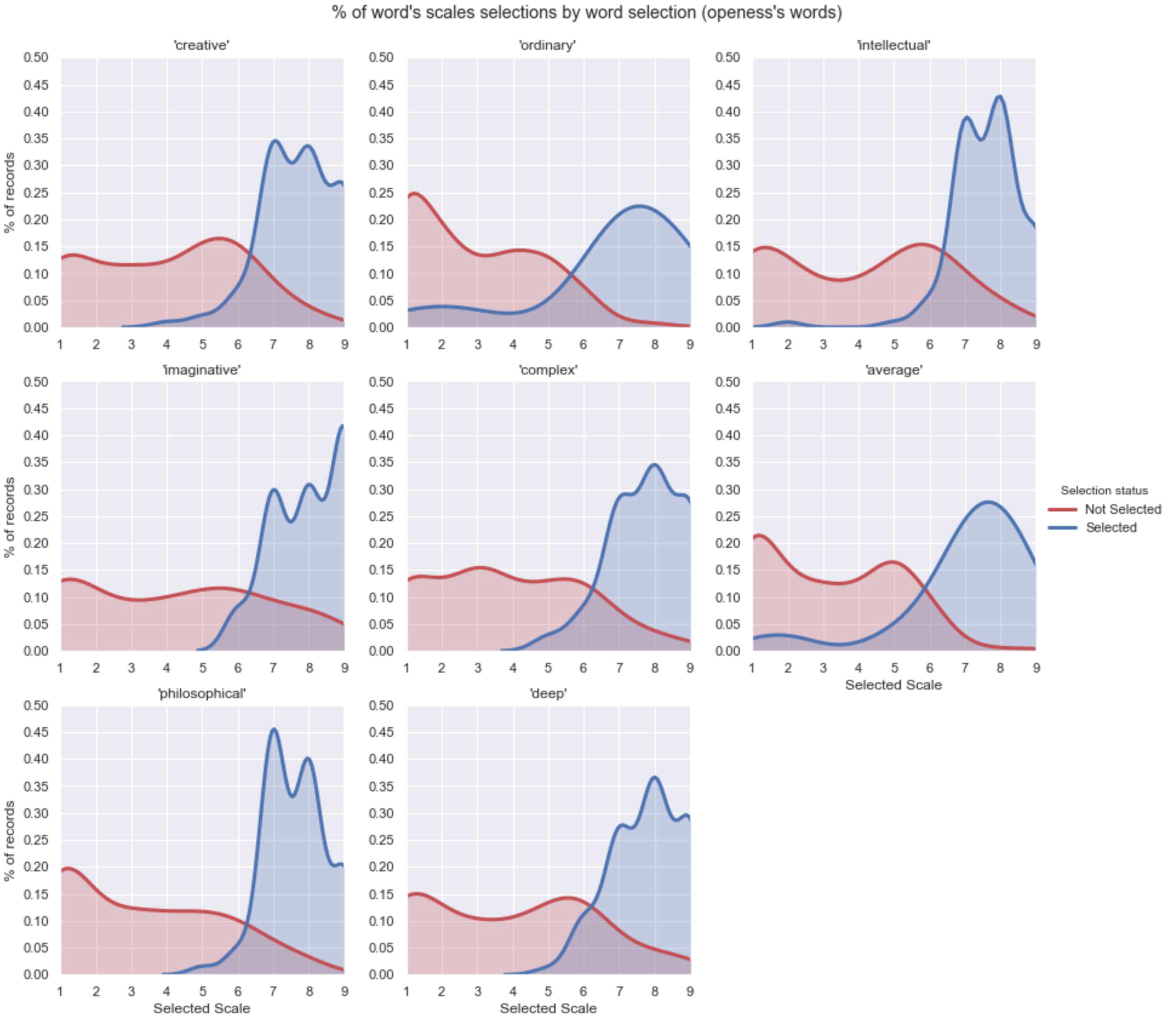
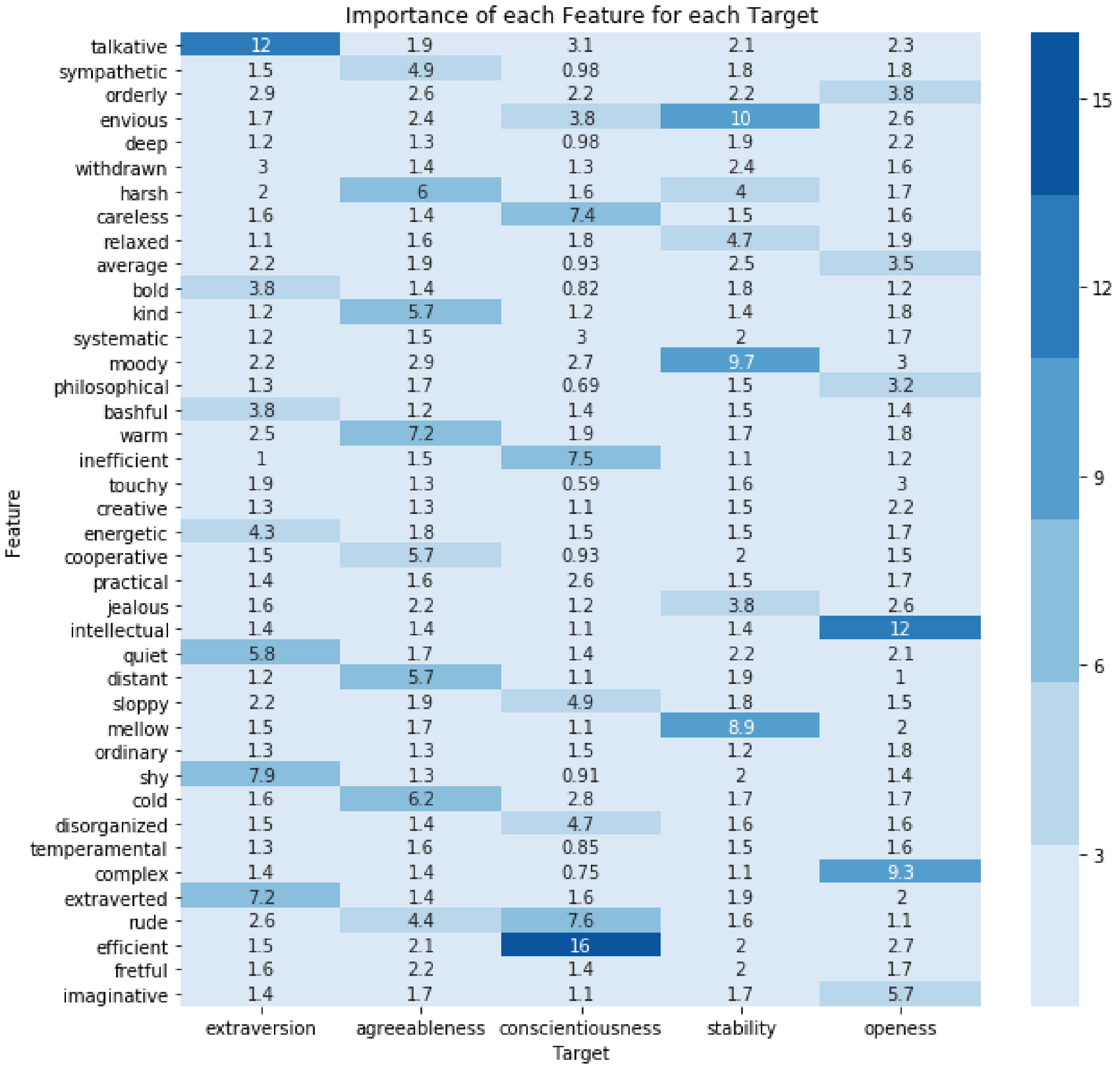


Figure 41.: Openness % of word's scales selections by word selection

## B.2 FEATURES IMPORTANCE

### B.2.1 *Approach I*



| Feature | extraversion | agreeableness | conscientiousness | stability | openess |
|---|---|---|---|---|---|
| talkative | 12 | 1.9 | 3.1 | 2.1 | 2.3 |
| sympathetic | 1.5 | 4.9 | 0.98 | 1.8 | 1.8 |
| orderly | 2.9 | 2.6 | 2.2 | 2.2 | 3.8 |
| envious | 1.7 | 2.4 | 3.8 | 10 | 2.6 |
| deep | 1.2 | 1.3 | 0.98 | 1.9 | 2.2 |
| withdrawn | 3 | 1.4 | 1.3 | 2.4 | 1.6 |
| harsh | 2 | 6 | 1.6 | 4 | 1.7 |
| careless | 1.6 | 1.4 | 7.4 | 1.5 | 1.6 |
| relaxed | 1.1 | 1.6 | 1.8 | 4.7 | 1.9 |
| average | 2.2 | 1.9 | 0.93 | 2.5 | 3.5 |
| bold | 3.8 | 1.4 | 0.82 | 1.8 | 1.2 |
| kind | 1.2 | 5.7 | 1.2 | 1.4 | 1.8 |
| systematic | 1.2 | 1.5 | 3 | 2 | 1.7 |
| moody | 2.2 | 2.9 | 2.7 | 9.7 | 3 |
| philosophical | 1.3 | 1.7 | 0.69 | 1.5 | 3.2 |
| bashful | 3.8 | 1.2 | 1.4 | 1.5 | 1.4 |
| warm | 2.5 | 7.2 | 1.9 | 1.7 | 1.8 |
| inefficient | 1 | 1.5 | 7.5 | 1.1 | 1.2 |
| touchy | 1.9 | 1.3 | 0.59 | 1.6 | 3 |
| creative | 1.3 | 1.3 | 1.1 | 1.5 | 2.2 |
| energetic | 4.3 | 1.8 | 1.5 | 1.5 | 1.7 |
| cooperative | 1.5 | 5.7 | 0.93 | 2 | 1.5 |
| practical | 1.4 | 1.6 | 2.6 | 1.5 | 1.7 |
| jealous | 1.6 | 2.2 | 1.2 | 3.8 | 2.6 |
| intellectual | 1.4 | 1.4 | 1.1 | 1.4 | 12 |
| quiet | 5.8 | 1.7 | 1.4 | 2.2 | 2.1 |
| distant | 1.2 | 5.7 | 1.1 | 1.9 | 1 |
| sloppy | 2.2 | 1.9 | 4.9 | 1.8 | 1.5 |
| mellow | 1.5 | 1.7 | 1.1 | 8.9 | 2 |
| ordinary | 1.3 | 1.3 | 1.5 | 1.2 | 1.8 |
| shy | 7.9 | 1.3 | 0.91 | 2 | 1.4 |
| cold | 1.6 | 6.2 | 2.8 | 1.7 | 1.7 |
| disorganized | 1.5 | 1.4 | 4.7 | 1.6 | 1.6 |
| temperamental | 1.3 | 1.6 | 0.85 | 1.5 | 1.6 |
| complex | 1.4 | 1.4 | 0.75 | 1.1 | 9.3 |
| extraverted | 7.2 | 1.4 | 1.6 | 1.9 | 2 |
| rude | 2.6 | 4.4 | 7.6 | 1.6 | 1.1 |
| efficient | 1.5 | 2.1 | 16 | 2 | 2.7 |
| fretful | 1.6 | 2.2 | 1.4 | 2 | 1.7 |
| imaginative | 1.4 | 1.7 | 1.1 | 1.7 | 5.7 |

Figure 42.: Features importance on Approach I with no Data Augmentation model

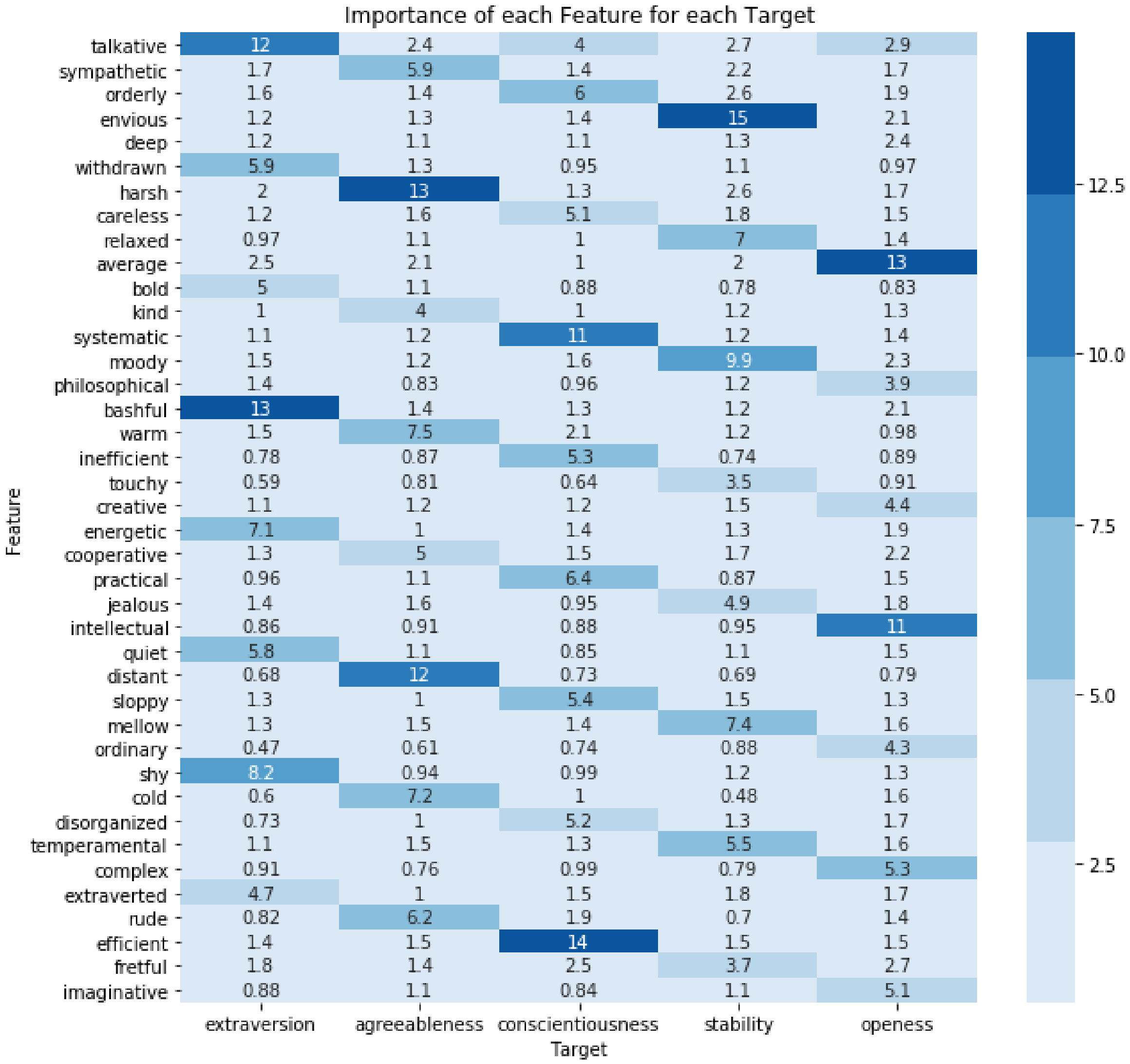


Figure 43.: Features importance on Approach I with Data Augmentation implementation I model

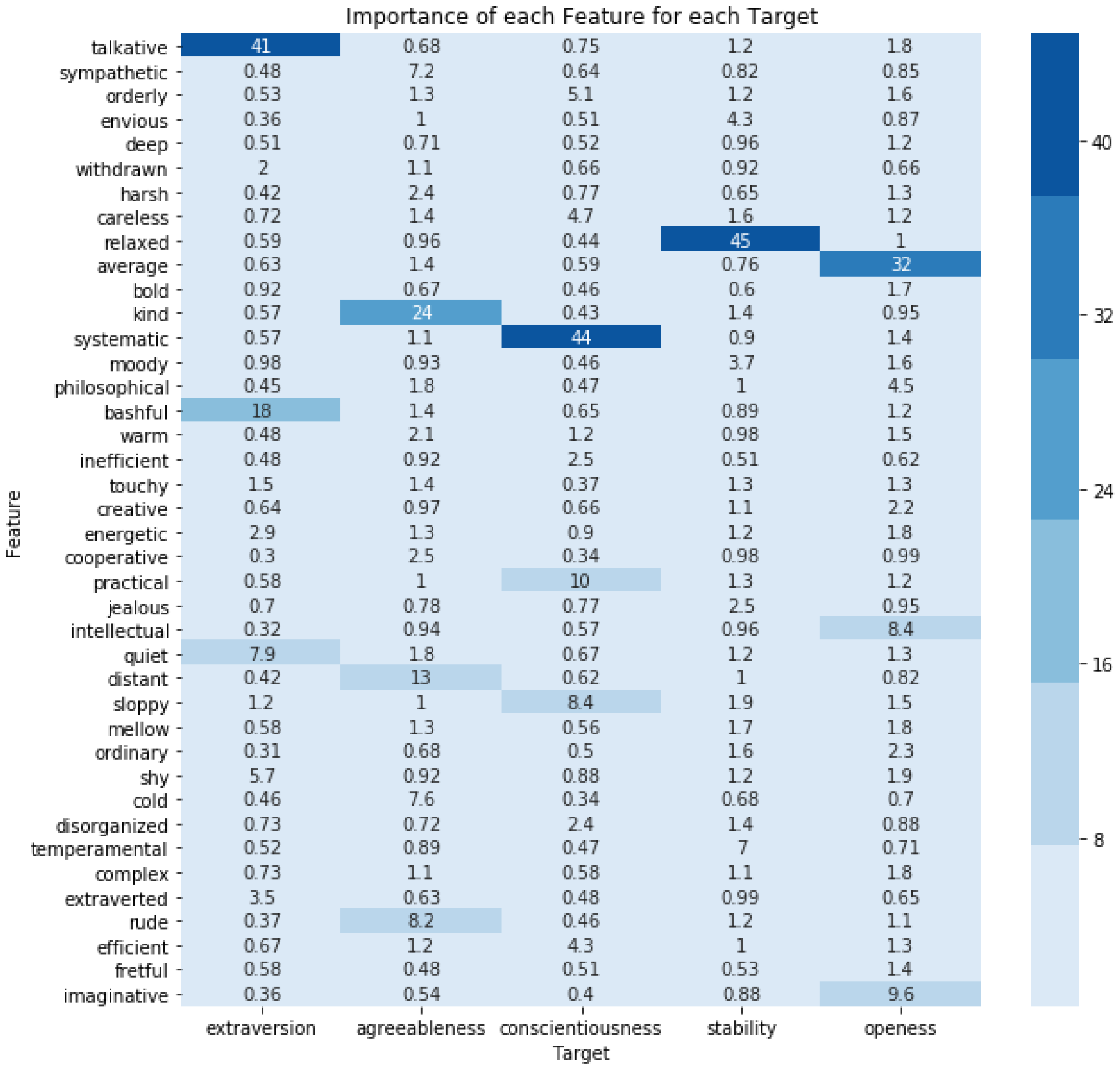


Figure 44.: Features importance on Approach I with Data Augmentation implementation II model

### B.2.2 *Approach II*

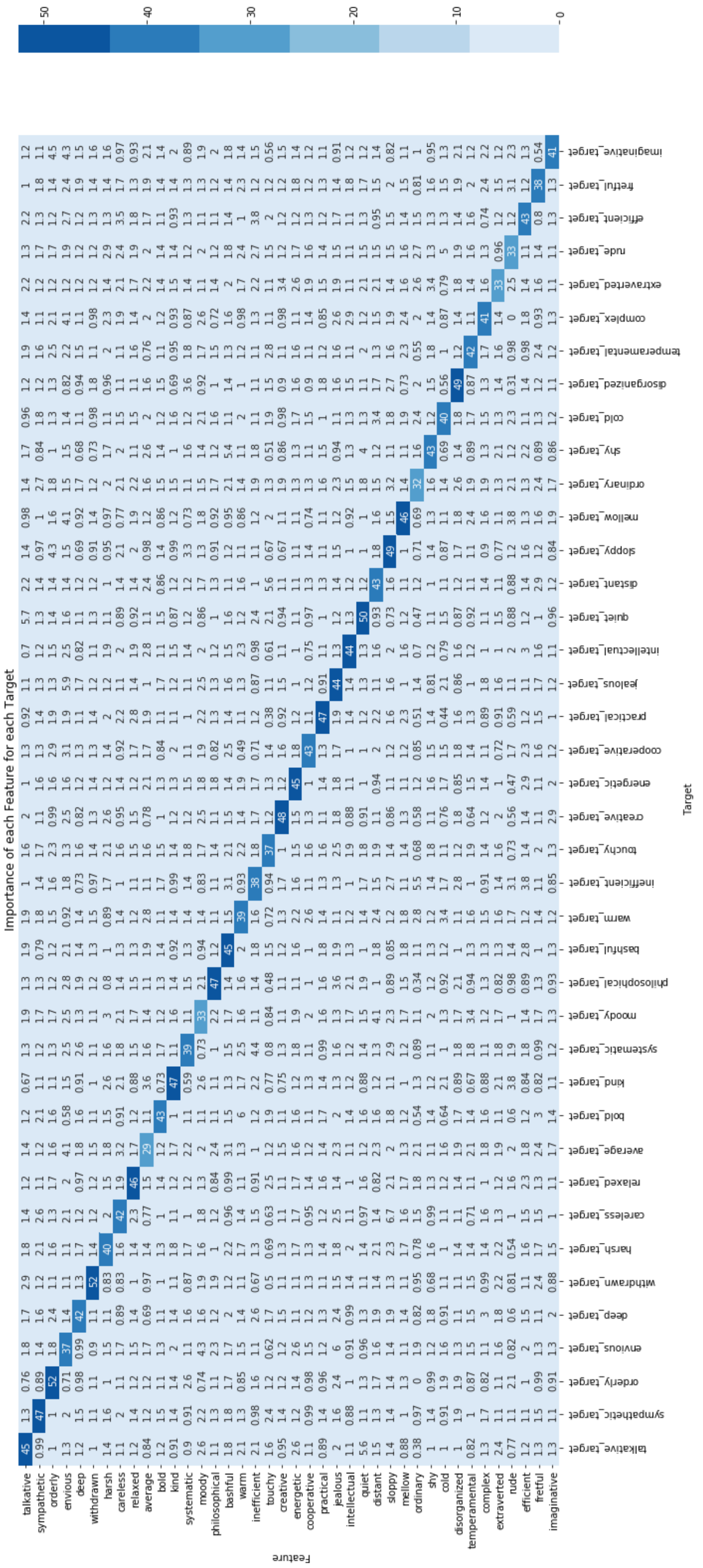


Figure 45.: Features importance on Approach II with no Data Augmentation model

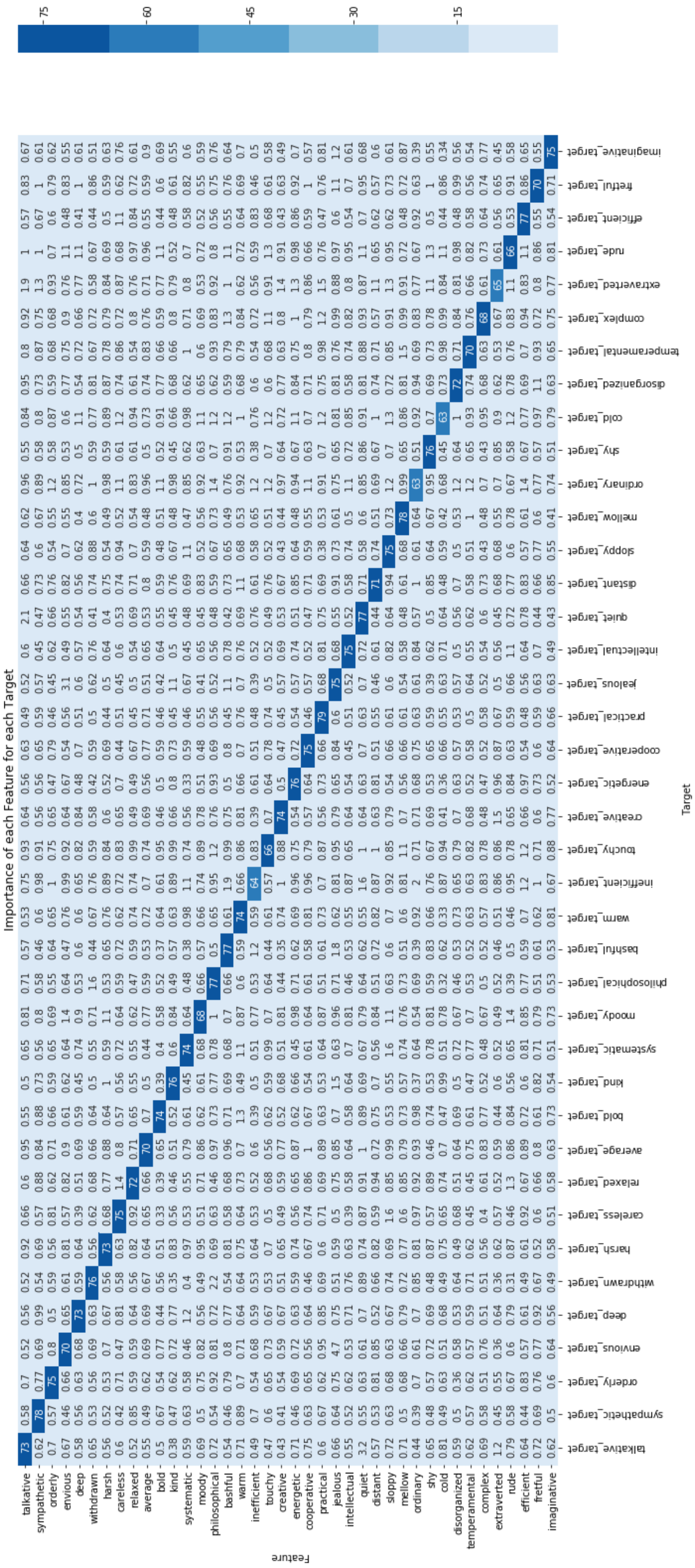


Figure 46.: Features importance on Approach II with Data Augmentation implementation I model

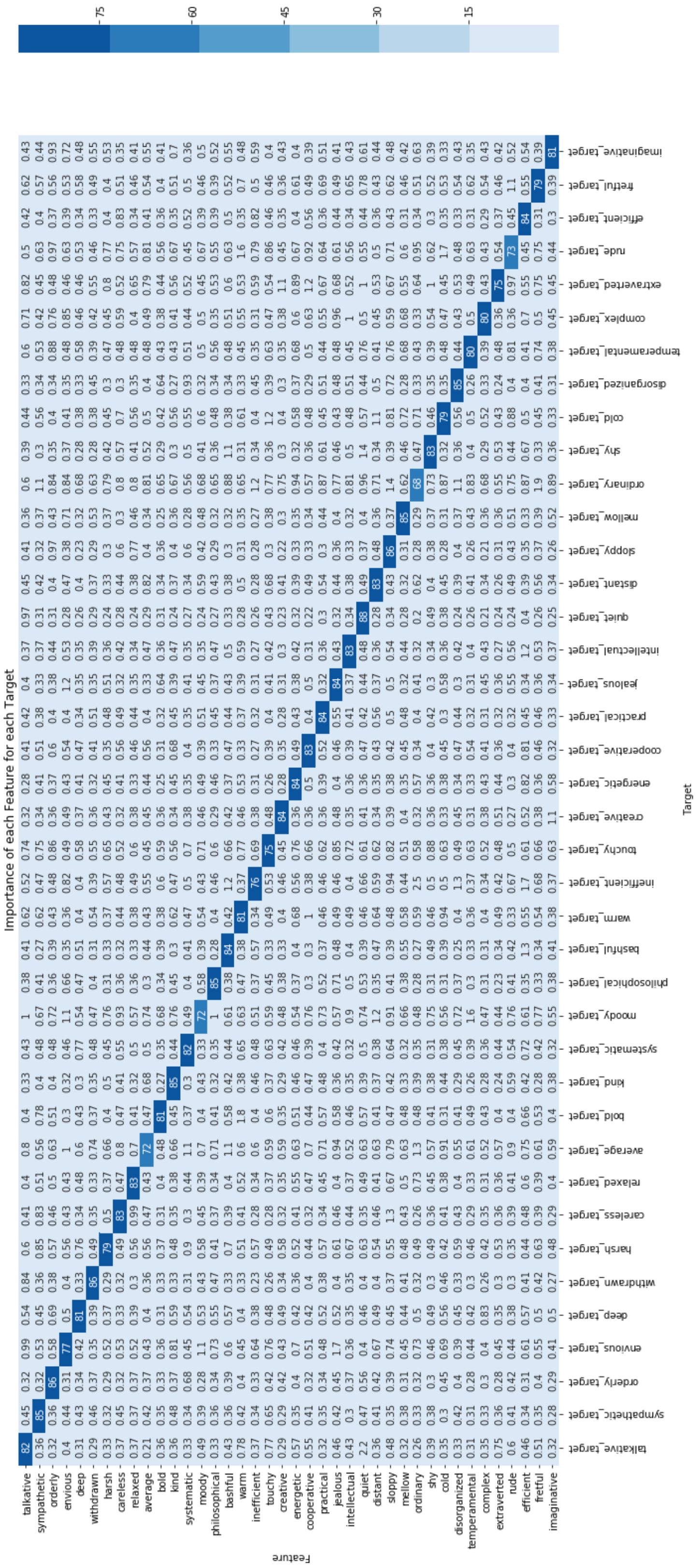


Figure 47.: Features importance on Approach II with Data Augmentation implementation II model

### B.2.3 *Approach III*

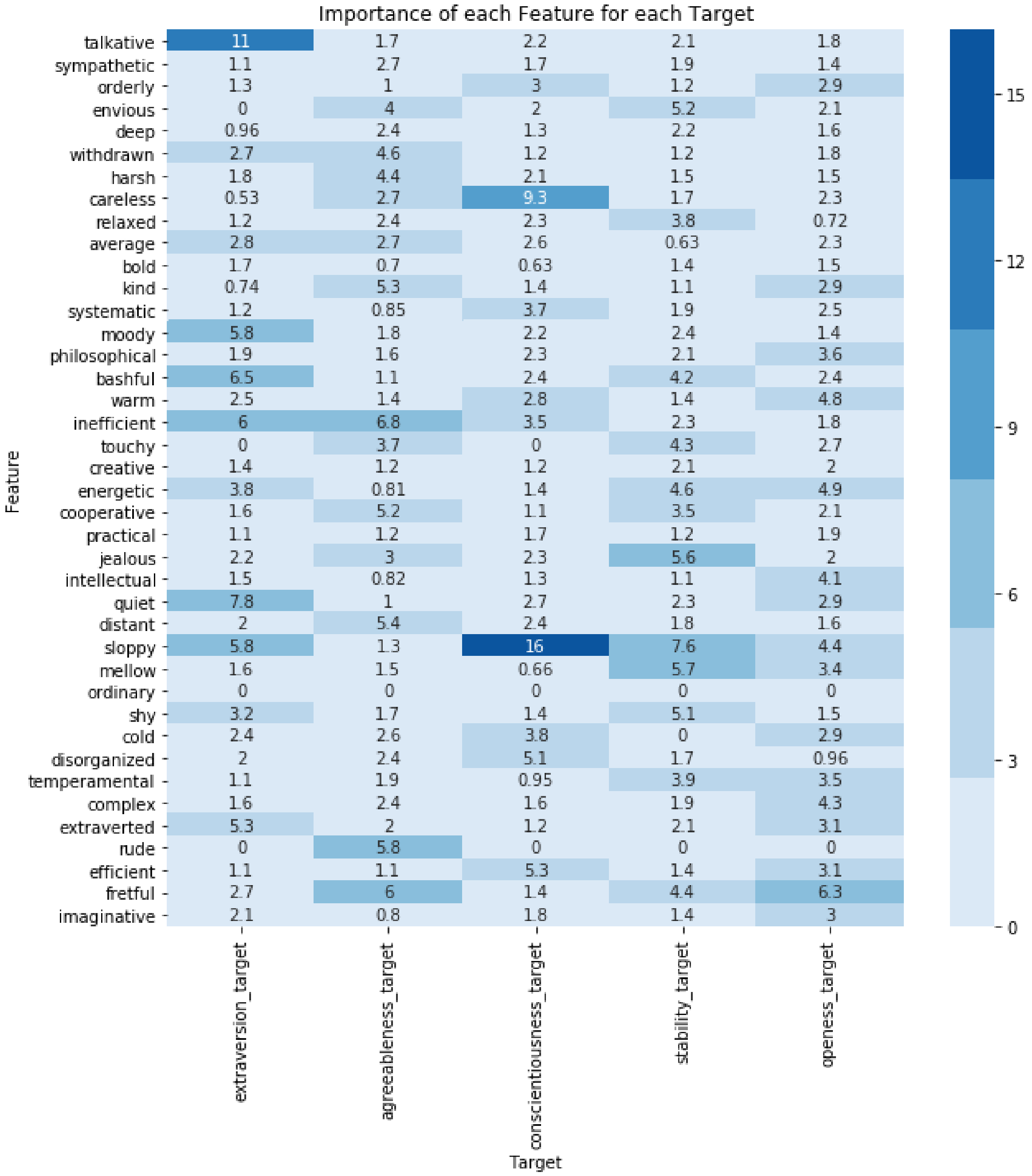


Figure 48.: Features importance on Approach III with no Data Augmentation model with 5 bins

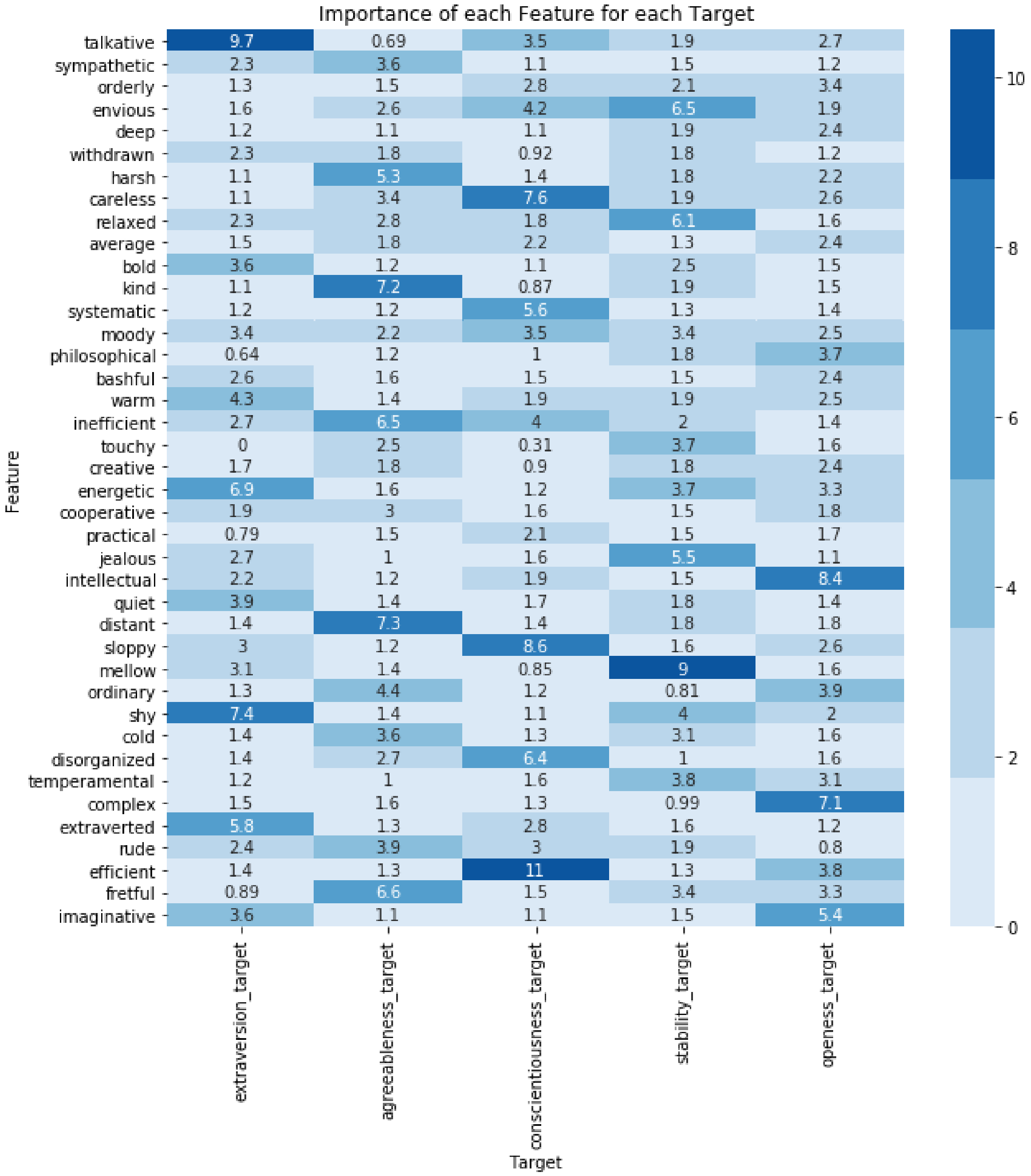


Figure 49.: Features importance on Approach III with no Data Augmentation model with 3 bins

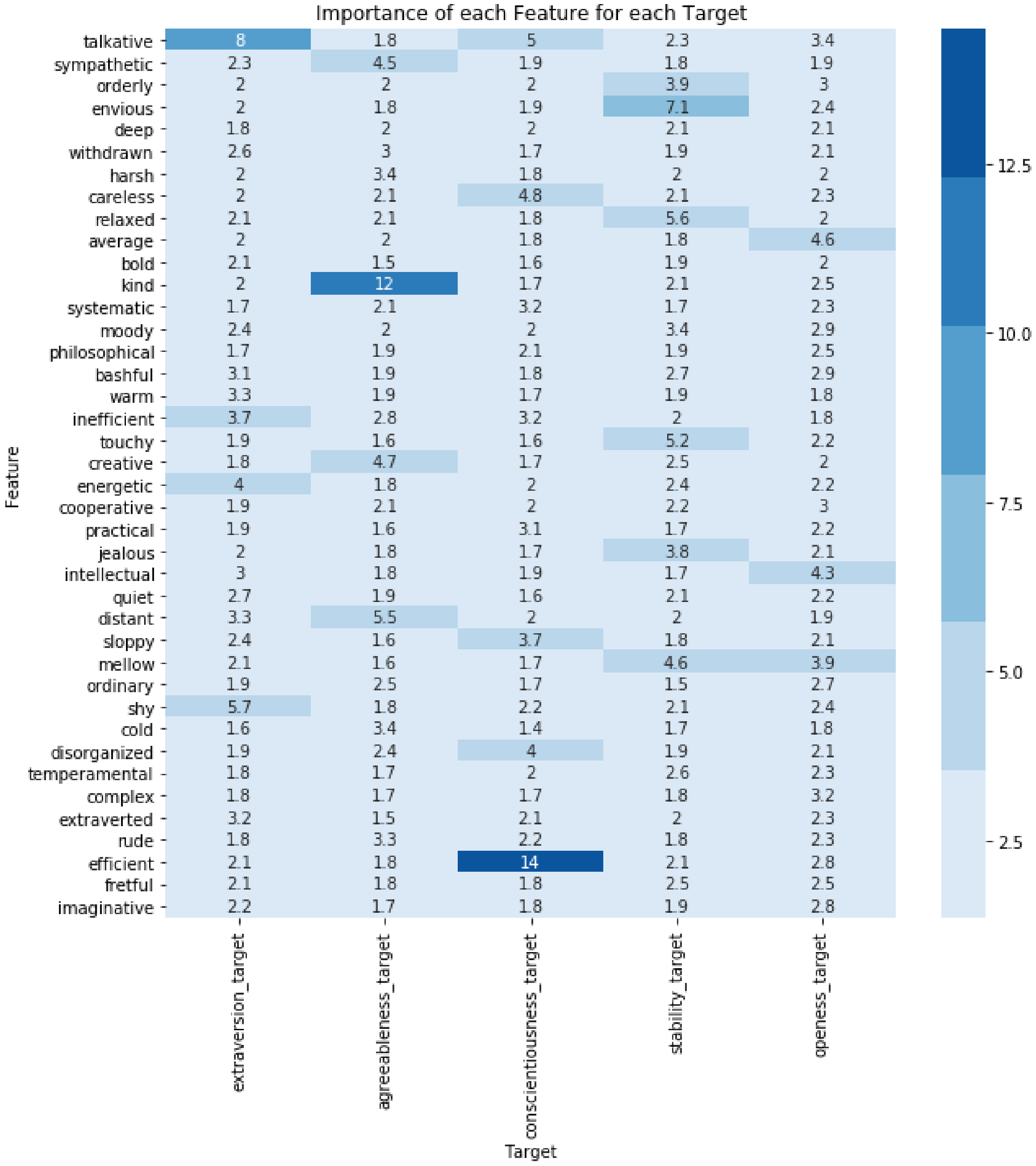


Figure 50.: Features importance on Approach III with Data Augmentation implementation I model with 3 bins

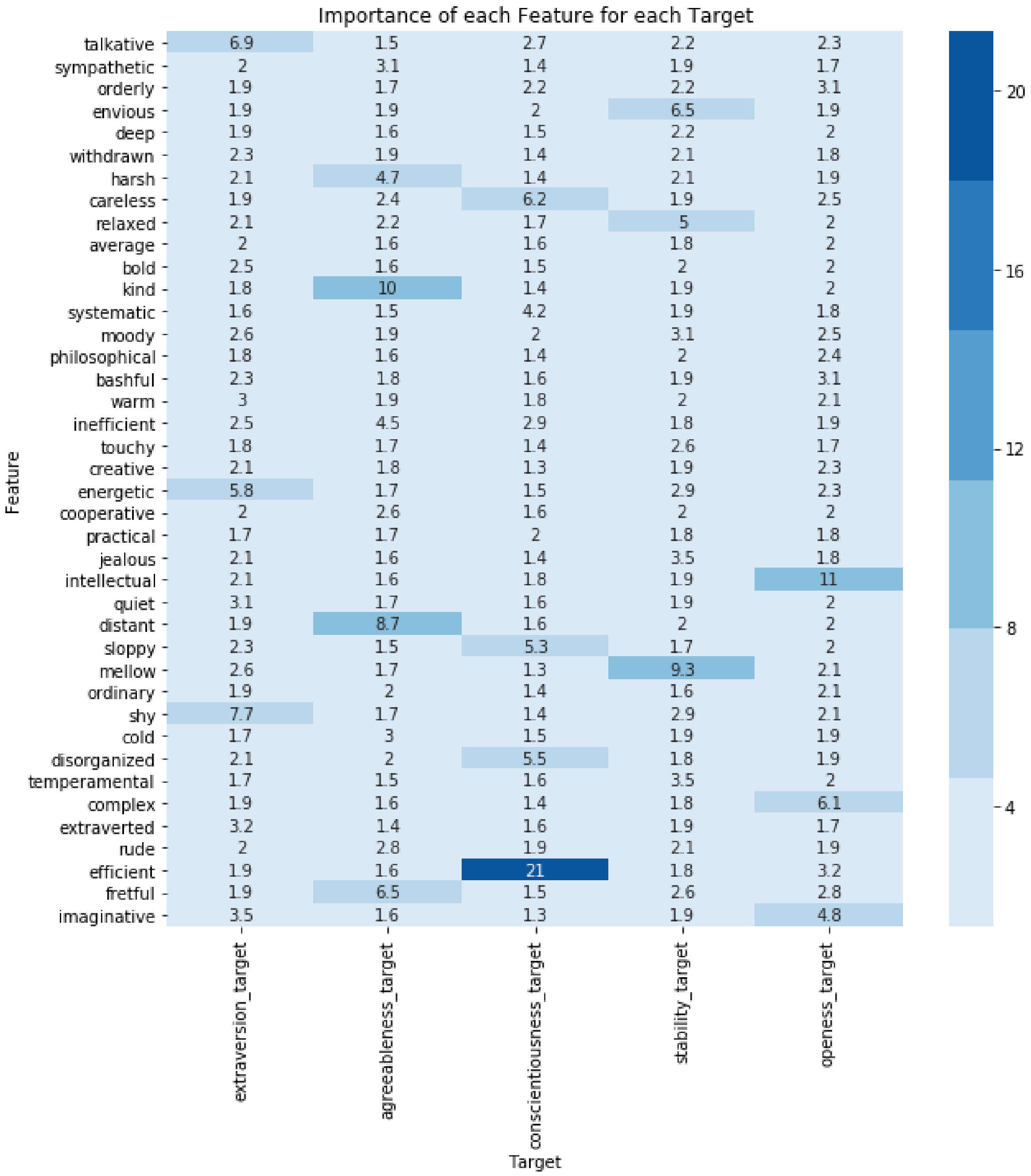


Figure 51.: Features importance on Approach III with Data Augmentation implementation II model with 3 bins

## B.3 PATTERN MINING

### B.3.1 *Support: 20*

[imaginative kind]

### B.3.2 *Support: 19*

[creative imaginative]

### B.3.3 *Support: 18*

[withdrawn intellectual]

### B.3.4 *Support: 17*

[intellectual kind]

### B.3.5 *Support: 16*

[withdrawn philosophical], [jealous kind], [deep kind], [creative orderly], [envious jealous], [sympathetic bold], [sympathetic kind]

### B.3.6 *Support: 15*

[imaginative mellow], [withdrawn sympathetic], [relaxed sympathetic], [imaginative jealous], [shy kind], [mellow kind], [energetic kind]

### B.3.7 *Support: 14*

[bashful jealous], [intellectual sympathetic], [bashful creative], [philosophical imaginative], [imaginative sympathetic], [shy bashful], [intellectual bold], [bashful kind], [withdrawn disorganized], [careless withdrawn], [envious imaginative], [withdrawn relaxed], [withdrawn bold], [withdrawn kind], [creative withdrawn], [withdrawn quiet], [withdrawn imaginative], [systematic kind], [careless relaxed], [careless kind], [careless imaginative], [imaginative orderly], [bold kind], [bashful bold], [bashful sympathetic], [bashful imaginative], [deep systematic], [sympathetic orderly], [shy creative], [intellectual imaginative], [shy practical], [creative mellow], [talkative kind]

### B.3.8 *Support: 13*

[creative harsh], [withdrawn systematic], [creative extraverted], [creative bold], [energetic practical], [intellectual practical], [bold orderly], [kind orderly], [quiet practical], [energetic efficient], [creative kind], [creative practical], [shy imaginative], [disorganized imaginative], [imaginative systematic], [creative quiet], [bashful envious], [creative sympathetic], [systematic sympathetic], [careless quiet], [withdrawn mellow], [shy quiet], [deep sympathetic], [shy sympathetic], [withdrawn envious]

### B.3.9 *Support: 12*

[relaxed kind], [envious bold], [talkative bold], [jealous bold], [relaxed imaginative], [careless creative], [creative relaxed], [disorganized bold], [imaginative bold], [intellectual orderly], [systematic efficient], [philosophical kind], [creative disorganized], [creative energetic], [bashful deep], [sympathetic practical], [bashful mellow], [bashful inefficient], [imaginative efficient], [withdrawn warm], [imaginative complex], [creative efficient], [sloppy kind], [intellectual energetic], [shy mellow], [imaginative energetic], [practical kind]

### B.3.10 *Support: 11*

[imaginative sloppy], [shy disorganized], [bashful jealous kind], [shy deep], [shy withdrawn], [intellectual efficient], [deep cooperative], [creative deep], [bashful withdrawn], [creative cooperative], [intellectual mellow], [quiet kind], [warm imaginative], [creative warm], [creative intellectual], [bashful quiet], [imaginative moody], [creative moody], [intellectual systematic], [warm systematic], [efficient kind], [cooperative kind], [careless cooperative], [warm kind], [cooperative practical], [bold practical], [energetic bold], [imaginative practical], [intellectual jealous], [bashful relaxed], [jealous mellow], [careless sloppy], [mellow bold], [talkative sympathetic], [envious energetic], [envious kind], [withdrawn orderly], [mellow orderly], [shy orderly], [disorganized orderly], [cooperative sympathetic], [envious orderly], [energetic talkative], [deep orderly], [careless deep], [intellectual talkative], [envious sympathetic], [extraverted talkative]

### B.3.11 *Support: 10*

[careless intellectual], [careless mellow], [envious moody], [relaxed practical], [shy relaxed], [relaxed efficient], [philosophical bold], [imaginative harsh], [systematic orderly], [creative envious], [extraverted kind], [envious harsh], [deep bold], [bashful systematic], [jealous systematic], [quiet energetic], [energetic cooperative], [warm sympathetic], [imaginative inefficient], [imaginative mellow kind], [warm energetic], [energetic sympathetic], [jealous sympathetic], [quiet mellow], [imaginative quiet], [imaginative distant], [disorganized mellow], [relaxed talkative], [envious imaginative jealous], [shy efficient], [envious intellectual], [quiet systematic], [shy systematic], [extraverted energetic], [withdrawn practical], [deep relaxed], [deep imaginative], [withdrawn cooperative], [deep energetic], [withdrawn energetic], [withdrawn relaxed sympathetic]

# C

# LISTINGS

## C.1 MODELLING LOG FILES

### C.1.1 *Approach I*

```
######## APPROACH I - No DA ########
========== path_training_set ==========
/data/processed/approach_i/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
3
========== mean_rs_score ==========
10.544753638403023
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=8, gamma=0.06,
colsample_bytree=0.3, eta=0.05,
n_estimators=500, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(182, 45)
========== mean_cv_score ==========
10.366321082521997
========== folds’ scores ==========
{
’MAE’: 8.873725853591669, ’MSE’: 156.0532476443462, ’RMSE’: 12.492127426677419
}, {
’MAE’: 6.038244003546042, ’MSE’: 56.888135187761534, ’RMSE’: 7.542422368693067
}, {
’MAE’: 7.514472961425781, ’MSE’: 109.31929321326785, ’RMSE’: 10.45558669866344
}, {
’MAE’: 8.154216715494792, ’MSE’: 120.45387002324223, ’RMSE’:
    10.975147836054065
}
```

Listing C.1: Approach I with no Data Augmentation

```
######## APPROACH I - DA I ########
========== path_training_set ==========
/data/processed/approach_i/augmented/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== mean_rs_score ==========
10.950405535232663
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=12,
min_child_weight=6, gamma=0.06,
colsample_bytree=0.4, eta=0.05,
n_estimators=800, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(4590, 45)
========== mean_cv_score ==========
10.31963550381931
========== folds’ scores ==========
{
’MAE’: 10.231932577494701, ’MSE’: 191.85043287218275, ’RMSE’:
    13.851008370230044
}, {
’MAE’: 5.0628875448189525, ’MSE’: 40.839155094209225, ’RMSE’:
    6.390552017956604
}, {
’MAE’: 5.2390917684517655, ’MSE’: 45.90854621032214, ’RMSE’:
    6.7755845659486935
}, {
’MAE’: 11.007101087881848, ’MSE’: 203.3874461355468, ’RMSE’: 14.2613970611419
}
```

Listing C.2: Approach I with Data Augmentation implementation I

```
######## APPROACH I - DA II ########
========== path_training_set ==========
/data/processed/approach_i/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== mean_rs_score ==========
6.780241554700738
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=6, gamma=0.06,
colsample_bytree=0.3, eta=0.15,
n_estimators=600, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(3075, 45)
========== mean_cv_score ==========
7.252199832708579
========== folds’ scores ==========
{
’MAE’: 6.570987319946289, ’MSE’: 65.56410882641308, ’RMSE’: 8.097166715982391
}, {
’MAE’: 5.346569580078126, ’MSE’: 44.74494421383366, ’RMSE’: 6.689166182255727
}, {
’MAE’: 5.570287872314454, ’MSE’: 49.282447671238444, ’RMSE’: 7.020145844014812
}, {
’MAE’: 5.810190444946289, ’MSE’: 51.873421860703296, ’RMSE’: 7.202320588581384
}
```

Listing C.3: Approach I with Data Augmentation implementation II

```
######## APPROACH I - DA III ########
========== path_training_set ==========
/data/processed/approach_i/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== mean_rs_score ==========
6.780241554700738
```

```
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=6, gamma=0.06,
colsample_bytree=0.3, eta=0.15,
n_estimators=600, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== pattern_proba ==========
0.6
========== Fold Dimensions ==========
(18943, 45)
========== Fold Dimensions ==========
(18857, 45)
========== Fold Dimensions ==========
(18966, 45)
========== Fold Dimensions ==========
(18938, 45)
========== mean_cv_score ==========
5.914032197011077
========== folds’ scores ==========
{
’MAE’: 5.34582487487793, ’MSE’: 43.84569427281947, ’RMSE’: 6.621608133438543
}, {
’MAE’: 4.283765838623047, ’MSE’: 30.19346269394079, ’RMSE’: 5.4948578411038795
}, {
’MAE’: 4.646395492553711, ’MSE’: 35.31491536298205, ’RMSE’: 5.942635388695999
}, {
’MAE’: 4.447961334228516, ’MSE’: 31.326715994029247, ’RMSE’: 5.59702742480589
}
```

Listing C.4: Approach I with Data Augmentation implementation III

### C.1.2 *Approach II*

```
######## APPROACH II - No DA ########
========== path_training_set ==========
/data/processed/approach_ii/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
3
========== mean_rs_score ==========
1.975170526843302
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=8, gamma=0.04,
colsample_bytree=0.5, eta=0.05,
n_estimators=500, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(182, 80)
========== mean_cv_score ==========
1.9461652735120125
========== folds’ scores ==========
{
’MAE’: 1.5900312847778448, ’MSE’: 3.7202643899111494, ’RMSE’:
    1.9287986908724168
}, {
’MAE’: 1.6708432238610065, ’MSE’: 4.097545908516227, ’RMSE’: 2.024239587725778
}, {
’MAE’: 1.5701171170981205, ’MSE’: 3.69999450322704, ’RMSE’: 1.9235369773485094
}, {
’MAE’: 1.5063545845697324, ’MSE’: 3.6407915655629184, ’RMSE’:
    1.9080858381013466
}
========== mean_cv_score (TRAITS) ==========
6.43713333291188
========== folds’ scores (TRAITS) ==========
{
’MAE’: 5.245901639344263, ’MSE’: 44.66229508196721, ’RMSE’: 6.682985491677145
}, {
’MAE’: 4.547540983606558, ’MSE’: 32.49508196721312, ’RMSE’: 5.700445769166927
}, {
’MAE’: 5.304918032786885, ’MSE’: 45.61311475409836, ’RMSE’: 6.753748200377207
}, {
```

```
'MAE': 4.956666666666667, 'MSE': 43.71, 'RMSE': 6.611353870426238
}
```

Listing C.5: Approach II with no Data Augmentation

```
######## APPROACH II - DA I ########
========== path_training_set ==========
/data/processed/approach_ii/augmented/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
'estimator__learning_rate': [0.001, 0.01],
'estimator__max_depth': [10, 12, 14],
'estimator__min_child_weight': [6, 7, 8],
'estimator__gamma': [0.04, 0.05, 0.06],
'estimator__colsample_bytree': [0.3, 0.4, 0.5],
'estimator__eta': [0.05, 0.1, 0.15],
'estimator__n_estimators': [500, 600, 700, 800], 'estimator__eval_metric': ['
    rmse'],
'estimator__objective': ['reg:squarederror']
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== mean_rs_score ==========
1.9643425165138801
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=8, gamma=0.04,
colsample_bytree=0.5, eta=0.05,
n_estimators=500, eval_metric='rmse',
objective='reg:squarederror'
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(4590, 80)
========== mean_cv_score ==========
1.962323474152395
========== folds' scores ==========
{
'MAE': 1.995897568989620, 'MSE': 3.807392502863189, 'RMSE': 1.1228079955949411
}, {
'MAE': 2.0621501535077699, 'MSE': 4.685997973850825, 'RMSE':
    1.0579455826314874
}, {
```

```
’MAE’: 1.86028599537758, ’MSE’: 3.875098039117817, ’RMSE’: 1.706722935889503
}, {
’MAE’: 1.93096017873469, ’MSE’: 3.297636508257768, ’RMSE’: 1.316953280221697
}
========== mean_cv_score (TRAITS) ==========
5.956814580560344
========== folds’ scores (TRAITS) ==========
{
’MAE’: 5.72883948277103000, ’MSE’: 37.934715734298527, ’RMSE’:
    5.431030957862271
}, {
’MAE’: 5.131653244111293, ’MSE’: 39.770019965076478, ’RMSE’: 6.389559600105887
}, {
’MAE’: 5.2401615314072338, ’MSE’: 40.0848385608760946, ’RMSE’:
    5.978348399774857
}, {
’MAE’: 5.2584262078272867, ’MSE’: 37.12718606811332012, ’RMSE’:
    6.02831936449836158
}
```

Listing C.6: Approach II with Data Augmentation implementation I

```
######## APPROACH II - DA II ########
========== path_training_set ==========
/data/processed/approach_ii/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== mean_rs_score ==========
1.9881983330112
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=8, gamma=0.04,
colsample_bytree=0.5, eta=0.05,
n_estimators=500, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(7462, 80)
========== mean_cv_score ==========
2.2145365850448018
========== folds’ scores ==========
{
’MAE’: 1.8619071773818283, ’MSE’: 4.807392502863189, ’RMSE’: 2.192576681182026
}, {
’MAE’: 2.015757410887812, ’MSE’: 5.685997973850825, ’RMSE’: 2.384533072500951
}, {
’MAE’: 1.8550124576834386, ’MSE’: 4.875098039117817, ’RMSE’: 2.207962417958652
}, {
’MAE’: 1.731705891912182, ’MSE’: 4.297636508257768, ’RMSE’: 2.0730741685375773
}
========== mean_cv_score (TRAITS) ==========
7.575203935401243
========== folds’ scores (TRAITS) ==========
{
’MAE’: 6.3213114754098365, ’MSE’: 64.68196721311475, ’RMSE’: 8.042510007026086
}, {
’MAE’: 5.501639344262296, ’MSE’: 48.0327868852459, ’RMSE’: 6.93056901597884
}, {
’MAE’: 6.327868852459017, ’MSE’: 62.68852459016394, ’RMSE’: 7.917608514580898
}, {
’MAE’: 5.7299999999999995, ’MSE’: 54.910000000000004, ’RMSE’:
   7.410128204019145
}
```

Listing C.7: Approach II with Data Augmentation implementation II

```
######## APPROACH II - DA III ########
========== path_training_set ==========
/data/processed/approach_ii/GENERAL.csv
========== Model ==========
XGBRegressor
========== metric_to_evaluate ==========
RMSE
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’rmse’],
’estimator__objective’: [’reg:squarederror’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== mean_rs_score ==========
1.9654688930133
========== Best hyperparameters ==========
learning_rate=0.01, max_depth=10,
min_child_weight=8, gamma=0.04,
colsample_bytree=0.5, eta=0.05,
n_estimators=500, eval_metric=’rmse’,
objective=’reg:squarederror’
========== total_cv_folds ==========
4
========== pattern_proba ==========
0.6
========== Fold Dimensions ==========
(18646, 80)
========== Fold Dimensions ==========
(19014, 80)
========== Fold Dimensions ==========
(18656, 80)
========== Fold Dimensions ==========
(19018, 80)
========== mean_cv_score ==========
1.83501901584904
========== folds’ scores ==========
{
’MAE’: 1.6001569962501527, ’MSE’: 3.5705789829782533, ’RMSE’:
    1.8895975717009834
}, {
’MAE’: 1.5538778927326202, ’MSE’: 3.5329676897807745, ’RMSE’:
    1.879619027830048
}, {
’MAE’: 1.5068769087791443, ’MSE’: 3.2680689063356256, ’RMSE’:
    1.807780104530312
}, {
’MAE’: 1.4824372777938843, ’MSE’: 3.1084488273124697, ’RMSE’:
    1.7630793593348173
}
========== mean_cv_score (TRAITS) ==========
5.896456172139214
========== folds' scores (TRAITS) ==========
{
'MAE': 5.4479999999999995, 'MSE': 44.74399999999999, 'RMSE': 6.689095604040952
}, {
'MAE': 4.136, 'MSE': 27.672000000000004, 'RMSE': 5.2604182343231995
}, {
'MAE': 4.656, 'MSE': 36.00000000000001, 'RMSE': 6.000000000000001
}, {
'MAE': 4.4719999999999995, 'MSE': 31.768, 'RMSE': 5.636310850192704
}
```

Listing C.8: Approach II with Data Augmentation implementation III

### c.1.3 *Approach III*

```
######## APPROACH III - No DA, 5 bins ########
========== path_training_set ==========
/data/processed/approach_iii/GENERAL.csv
========== Num Bins ==========
5
========== Model ==========
XGBClassifier(multi:softmax)
========== metric_to_evaluate ==========
F1
========== param_grid ==========
'estimator__learning_rate': [0.001, 0.01],
'estimator__max_depth': [10, 12, 14],
'estimator__min_child_weight': [6, 7, 8],
'estimator__gamma': [0.04, 0.05, 0.06],
'estimator__colsample_bytree': [0.3, 0.4, 0.5],
'estimator__eta': [0.05, 0.1, 0.15],
'estimator__n_estimators': [500, 600, 700, 800],
'estimator__eval_metric': ['auc']
========== Total combinations ==========
1944
========== total_rs_folds ==========
3
========== Best hyperparameters ==========
learning_rate=0.001, max_depth=14,
min_child_weight=6, gamma=0.05,
colsample_bytree=0.5, eta=0.1,
n_estimators=700, eval_metric='auc'
========== total_cv_folds ==========
4
```

```
========== Fold Dimensions ==========
(182, 45)
========== mean_cv_score ==========
’f1’: 0.4025136612021858,
’merror’: 0.5974863387978142
========== folds’ scores ==========
{
’merror’: 0.6327868852459017,
’f1_class’: [0.5245901639344263, 0.32786885245901637, 0.32786885245901637,
   0.3606557377049181, 0.29508196721311475],
’precision_class’: [0.5245901639344263, 0.32786885245901637,
   0.32786885245901637, 0.36065573770491804, 0.29508196721311475],
’recall_class’: [0.5245901639344263, 0.32786885245901637, 0.32786885245901637,
    0.36065573770491804, 0.29508196721311475],
’f1’: 0.36721311475409835,
’precision’: 0.36721311475409835,
’recall’: 0.36721311475409835
}, {
’merror’: 0.5442622950819672,
’f1_class’: [0.639344262295082, 0.4098360655737705, 0.47540983606557374,
   0.4262295081967213, 0.32786885245901637],
’precision_class’: [0.639344262295082, 0.4098360655737705,
   0.47540983606557374, 0.4262295081967213, 0.32786885245901637],
’recall_class’: [0.639344262295082, 0.4098360655737705, 0.47540983606557374,
   0.4262295081967213, 0.32786885245901637],
’f1’: 0.4557377049180328,
’precision’: 0.4557377049180328,
’recall’: 0.4557377049180328
}, {
’merror’: 0.6262295081967213,
’f1_class’: [0.3770491803278688, 0.3606557377049181, 0.32786885245901637,
   0.4262295081967213, 0.3770491803278688],
’precision_class’: [0.3770491803278688, 0.36065573770491804,
   0.32786885245901637, 0.4262295081967213, 0.3770491803278688],
’recall_class’: [0.3770491803278688, 0.36065573770491804, 0.32786885245901637,
    0.4262295081967213, 0.3770491803278688],
’f1’: 0.3737704918032787,
’precision’: 0.37377049180327865,
’recall’: 0.37377049180327865
}, {
’merror’: 0.5866666666666667,
’f1_class’: [0.4666666666666667, 0.36666666666666664, 0.43333333333333335,
   0.38333333333333336, 0.4166666666666667],
’precision_class’: [0.4666666666666667, 0.36666666666666664,
   0.43333333333333335, 0.38333333333333336, 0.4166666666666667],
’recall_class’: [0.4666666666666667, 0.36666666666666664, 0.43333333333333335,
    0.38333333333333336, 0.4166666666666667],
'f1': 0.4133333333333333,
'precision': 0.4133333333333333,
'recall': 0.4133333333333333
}
```

Listing C.9: Approach III with no Data Augmentation with 5 bins

```
######## APPROACH III - No DA ########
========== path_training_set ==========
/data/processed/approach_iii/GENERAL.csv
========== Num Bins ==========
3
========== Model ==========
XGBClassifier(multi:softmax)
========== metric_to_evaluate ==========
F1
========== param_grid ==========
'estimator__learning_rate': [0.001, 0.01],
'estimator__max_depth': [10, 12, 14],
'estimator__min_child_weight': [6, 7, 8],
'estimator__gamma': [0.04, 0.05, 0.06],
'estimator__colsample_bytree': [0.3, 0.4, 0.5],
'estimator__eta': [0.05, 0.1, 0.15],
'estimator__n_estimators': [500, 600, 700, 800],
'estimator__eval_metric': ['auc']
========== Total combinations ==========
1944
========== total_rs_folds ==========
3
========== Best hyperparameters ==========
learning_rate=0.001, max_depth=12,
min_child_weight=7, gamma=0.05,
colsample_bytree=0.3, eta=0.1,
n_estimators=800, eval_metric='auc'
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(182, 45)
========== mean_cv_score ==========
'f1': 0.6400409836065575,
'merror': 0.3599590163934426
========== folds' scores ==========
{
'merror': 0.3901639344262295,
'f1_class': [0.6721311475409836, 0.6065573770491803, 0.6721311475409836,
    0.5737704918032787, 0.5245901639344263],
’precision_class’: [0.6721311475409836, 0.6065573770491803,
    0.6721311475409836, 0.5737704918032787, 0.5245901639344263],
’recall_class’: [0.6721311475409836, 0.6065573770491803, 0.6721311475409836,
    0.5737704918032787, 0.5245901639344263],
’f1’: 0.6098360655737706,
’precision’: 0.6098360655737706,
’recall’: 0.6098360655737706
}, {
’merror’: 0.2262295081967213,
’f1_class’: [0.819672131147541, 0.7213114754098362, 0.7868852459016392,
    0.7377049180327869, 0.8032786885245902],
’precision_class’: [0.819672131147541, 0.7213114754098361, 0.7868852459016393,
     0.7377049180327869, 0.8032786885245902],
’recall_class’: [0.819672131147541, 0.7213114754098361, 0.7868852459016393,
    0.7377049180327869, 0.8032786885245902],
’f1’: 0.7737704918032786,
’precision’: 0.7737704918032786,
’recall’: 0.7737704918032786
}, {
’merror’: 0.39344262295081966,
’f1_class’: [0.6557377049180327, 0.5737704918032787, 0.5573770491803278,
    0.5901639344262295, 0.6557377049180327],
’precision_class’: [0.6557377049180327, 0.5737704918032787,
    0.5573770491803278, 0.5901639344262295, 0.6557377049180327],
’recall_class’: [0.6557377049180327, 0.5737704918032787, 0.5573770491803278,
    0.5901639344262295, 0.6557377049180327],
’f1’: 0.6065573770491802,
’precision’: 0.6065573770491802,
’recall’: 0.6065573770491802
}, {
’merror’: 0.43,
’f1_class’: [0.55, 0.5166666666666667, 0.6166666666666667, 0.6166666666666667,
     0.55],
’precision_class’: [0.55, 0.5166666666666667, 0.6166666666666667,
    0.6166666666666667, 0.55],
’recall_class’: [0.55, 0.5166666666666667, 0.6166666666666667,
    0.6166666666666667, 0.55],
’f1’: 0.5700000000000001,
’precision’: 0.5700000000000001,
’recall’: 0.5700000000000001
}
```

Listing C.10: Approach III with no Data Augmentation with 3 bins

```
######## APPROACH III - DA I ########
========== path_training_set ==========
/data/processed/approach_iii/augmented/GENERAL.csv
```

```
========== Num Bins ==========
3
========== Model ==========
XGBClassifier
========== metric_to_evaluate ==========
F1
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’auc’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== Best hyperparameters ==========
learning_rate=0.001, max_depth=12,
min_child_weight=7, gamma=0.05,
colsample_bytree=0.3, eta=0.1,
n_estimators=800, eval_metric=’auc’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(4590, 45)
========== mean_cv_score ==========
’f1’: 0.642483660130719,
’merror’: 0.357516339869281
========== folds’ scores ==========
{
’merror’: 0.4373856209150327,
’f1_class’: [0.6169934640522876, 0.6189542483660131, 0.4738562091503268,
    0.5843137254901961, 0.5189542483660131],
’precision_class’: [0.6169934640522876, 0.6189542483660131,
    0.4738562091503268, 0.5843137254901961, 0.5189542483660131],
’recall_class’: [0.6169934640522876, 0.6189542483660131, 0.4738562091503268,
    0.5843137254901961, 0.5189542483660131],
’f1’: 0.5626143790849674,
’precision’: 0.5626143790849674,
’recall’: 0.5626143790849674
}, {
’merror’: 0.20313725490196077,
’f1_class’: [0.7980392156862746, 0.7980392156862746, 0.7816993464052286,
    0.8437908496732026, 0.7627450980392156],
'precision_class': [0.7980392156862746, 0.7980392156862746,
    0.7816993464052288, 0.8437908496732026, 0.7627450980392156],
'recall_class': [0.7980392156862746, 0.7980392156862746, 0.7816993464052288,
    0.8437908496732026, 0.7627450980392156],
'f1': 0.7968627450980392,
'precision': 0.7968627450980392,
'recall': 0.7968627450980392
}, {
'merror': 0.25058823529411767,
'f1_class': [0.7666666666666667, 0.7633986928104574, 0.738562091503268,
    0.677124183006536, 0.8013071895424837],
'precision_class': [0.7666666666666667, 0.7633986928104575, 0.738562091503268,
     0.677124183006536, 0.8013071895424837],
'recall_class': [0.7666666666666667, 0.7633986928104575, 0.738562091503268,
    0.677124183006536, 0.8013071895424837],
'f1': 0.7494117647058823,
'precision': 0.7494117647058823,
'recall': 0.7494117647058823
}, {
'merror': 0.538954248366013,
'f1_class': [0.4647058823529412, 0.4137254901960784, 0.4477124183006536,
    0.4934640522875817, 0.48562091503267973],
'precision_class': [0.4647058823529412, 0.4137254901960784,
    0.4477124183006536, 0.4934640522875817, 0.48562091503267973],
'recall_class': [0.4647058823529412, 0.4137254901960784, 0.4477124183006536,
    0.4934640522875817, 0.48562091503267973],
'f1': 0.46104575163398687,
'precision': 0.46104575163398687,
'recall': 0.46104575163398687
}
```

Listing C.11: Approach III with Data Augmentation implementation I

```
######## APPROACH III - DA II ########
========== path_training_set ==========
/data/processed/approach_iii/GENERAL.csv
========== Model ==========
XGBClassifier
========== metric_to_evaluate ==========
F1
========== param_grid ==========
'estimator__learning_rate': [0.001, 0.01],
'estimator__max_depth': [10, 12, 14],
'estimator__min_child_weight': [6, 7, 8],
'estimator__gamma': [0.04, 0.05, 0.06],
'estimator__colsample_bytree': [0.3, 0.4, 0.5],
'estimator__eta': [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’auc’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== Best hyperparameters ==========
learning_rate=0.001, max_depth=12,
min_child_weight=7, gamma=0.05,
colsample_bytree=0.3, eta=0.1,
n_estimators=800, eval_metric=’auc’
========== total_cv_folds ==========
4
========== Fold Dimensions ==========
(7462, 45)
========== mean_cv_score ==========
’f1’: 0.6409153005464481,
’merror’: 0.3590846994535519
========== folds’ scores ==========
{
’merror’: 0.4,
’f1_class’: [0.6557377049180327, 0.639344262295082, 0.5245901639344263,
   0.6229508196721312, 0.5573770491803278],
’precision_class’: [0.6557377049180327, 0.639344262295082, 0.5245901639344263,
    0.6229508196721312, 0.5573770491803278],
’recall_class’: [0.6557377049180327, 0.639344262295082, 0.5245901639344263,
   0.6229508196721312, 0.5573770491803278],
’f1’: 0.6,
’precision’: 0.6,
’recall’: 0.6
}, {
’merror’: 0.2459016393442623,
’f1_class’: [0.8524590163934426, 0.7377049180327869, 0.6557377049180327,
   0.7377049180327869, 0.7868852459016392],
’precision_class’: [0.8524590163934426, 0.7377049180327869,
   0.6557377049180327, 0.7377049180327869, 0.7868852459016393],
’recall_class’: [0.8524590163934426, 0.7377049180327869, 0.6557377049180327,
   0.7377049180327869, 0.7868852459016393],
’f1’: 0.7540983606557378,
’precision’: 0.7540983606557378,
’recall’: 0.7540983606557378
}, {
’merror’: 0.3737704918032787,
’f1_class’: [0.6557377049180327, 0.6229508196721312, 0.5409836065573771,
   0.639344262295082, 0.6721311475409836],
’precision_class’: [0.6557377049180327, 0.6229508196721312,
   0.5409836065573771, 0.639344262295082, 0.6721311475409836],
’recall_class’: [0.6557377049180327, 0.6229508196721312, 0.5409836065573771,
    0.639344262295082, 0.6721311475409836],
’f1’: 0.6262295081967213,
’precision’: 0.6262295081967213,
’recall’: 0.6262295081967213
}, {
’merror’: 0.4166666666666667,
’f1_class’: [0.5333333333333333, 0.5166666666666667, 0.6, 0.6166666666666667,
    0.65],
’precision_class’: [0.5333333333333333, 0.5166666666666667, 0.6,
    0.6166666666666667, 0.65],
’recall_class’: [0.5333333333333333, 0.5166666666666667, 0.6,
    0.6166666666666667, 0.65],
’f1’: 0.5833333333333333,
’precision’: 0.5833333333333333,
’recall’: 0.5833333333333333
}
```

Listing C.12: Approach III with Data Augmentation implementation II with 3 bins

```
######## APPROACH III - DA III ########
========== path_training_set ==========
/data/processed/approach_iii/GENERAL.csv
========== Model ==========
XGBClassifier
========== metric_to_evaluate ==========
F1
========== param_grid ==========
’estimator__learning_rate’: [0.001, 0.01],
’estimator__max_depth’: [10, 12, 14],
’estimator__min_child_weight’: [6, 7, 8],
’estimator__gamma’: [0.04, 0.05, 0.06],
’estimator__colsample_bytree’: [0.3, 0.4, 0.5],
’estimator__eta’: [0.05, 0.1, 0.15],
’estimator__n_estimators’: [500, 600, 700, 800],
’estimator__eval_metric’: [’auc’]
========== Total combinations ==========
1944
========== total_rs_folds ==========
2
========== Best hyperparameters ==========
learning_rate=0.001, max_depth=12,
min_child_weight=7, gamma=0.05,
colsample_bytree=0.3, eta=0.1,
n_estimators=800, eval_metric=’auc’
========== total_cv_folds ==========
4
```

```
========== pattern_proba ==========
0.6
========== Fold Dimensions ==========
(18936, 45)
========== Fold Dimensions ==========
(18645, 45)
========== Fold Dimensions ==========
(18787, 45)
========== Fold Dimensions ==========
(18956, 45)
========== mean_cv_score ==========
’f1’: 0.7420000000000001,
’merror’: 0.258
========== metrics ==========
{
’merror’: 0.272,
’f1_class’: [0.8000000000000002, 0.72, 0.68, 0.72, 0.72],
’precision_class’: [0.8, 0.72, 0.68, 0.72, 0.72],
’recall_class’: [0.8, 0.72, 0.68, 0.72, 0.72],
’f1’: 0.728,
’precision’: 0.728,
’recall’: 0.728
}, {
’merror’: 0.28,
’f1_class’: [0.72, 0.8000000000000002, 0.68, 0.72, 0.68],
’precision_class’: [0.72, 0.8, 0.68, 0.72, 0.68],
’recall_class’: [0.72, 0.8, 0.68, 0.72, 0.68],
’f1’: 0.72,
’precision’: 0.72,
’recall’: 0.72
}, {
’merror’: 0.208,
’f1_class’: [0.88, 0.68, 0.76, 0.76, 0.88],
’precision_class’: [0.88, 0.68, 0.76, 0.76, 0.88],
’recall_class’: [0.88, 0.68, 0.76, 0.76, 0.88],
’f1’: 0.792,
’precision’: 0.792,
’recall’: 0.792
}, {
’merror’: 0.272,
’f1_class’: [0.72, 0.8000000000000002, 0.72, 0.64, 0.76],
’precision_class’: [0.72, 0.8, 0.72, 0.64, 0.76],
’recall_class’: [0.72, 0.8, 0.72, 0.64, 0.76],
’f1’: 0.7280000000000001,
’precision’: 0.7280000000000001,
’recall’: 0.7280000000000001
}
```

Listing C.13: Approach III with Data Augmentation implementation III with 3 bins