ML Case Studies

Contents

Preface

This book is the result of a student projects for Case Studies[1] course at the Warsaw University of Technology. Each team prepared an article on one of the topics selected from reproducibility, imputation, and interpretability.

This project is inspired by a book Limitations of Interpretable Machine Learning Methods[2] created at the Department of Statistics, LMU Munich XAI Stories. Case studies for eXplainable Artificial Intelligence[3] done at the Warsaw University of Technology and at the University of Warsaw. We used the LIML project as the cornerstone for this repository.

The cover created by Anna Kozak.

FIGURE 1: Creative Commons License

[1] https://github.com/mini-pw/2020L-WarsztatyBadawcze
[2] https://compstat-lmu.github.io/iml_methods_limitations/
[3] https://pbiecek.github.io/xai_stories/
[4] http://creativecommons.org/licenses/by-nc-sa/4.0/

Technical Setup

The book chapters are written in the Markdown language. The simulations, data examples and visualizations were created with R (**?**) and Python. The book was compiled with the bookdown package. We collaborated using git and github. For details, head over to the book's repository[5].

[5]https://github.com/mini-pw/2020L-WB-Book

1 Reproducibility of scientific papers

The analysis of reproducibility of tools-related scientific papers.

1.1 Title of the article

Authors: Author 1, Author 2, Author 3 (University)

1.1.1 Abstract

1.1.2 Introduction and Motivation

1.1.3 Related Work

1.1.4 Methodology

1.1.5 Results

1.1.6 Summary and conclusions

1.2 How to measure reproducibility? Classification of problems with reproducing scientific papers

Authors: Paweł Koźmiński, Anna Urbala, Wojciech Szczypek (Warsaw University of Technology)

1.2.1 Abstract

1.2.2 Introduction

The idea of reproducibility of scientific researches is crucial especially in the area of data science. It has become more important along with the development of methods and algorithms used in machine learning as they are more and more complex and complicated. This issue concerns users of all types: students, scientists, developers. Moreover, attaching code used in a paper, helps readers to focus on the real content rather than sophisticated explanations and descriptions included in the article. It is also valuable because the users can use the code as examples of using the package.

However problem of the reproducibility is much more complex, because there is no explicit way of measuring it. It means that most of its definitions divide articles into 2 groups - reproducible and irreproducible. Thus, finding an appropriate reproducibility metrics, which would have wider set of values would result in changing the way reproducability is perceived. As a result such a metric would provide much more information for a person who would be interested in reproducing an article.

1.2.2.1 Definition

Reproducibility as a problem has been addressed by scientists of various fields of studies. The exact definition also differs among areas of studies. For instance, Patrick Vandewall in 2009 suggested a definition of a reproducible research work: "A research work is called reproducible if all information relevant to the work, including, but not limited to, text, data and code, is made available, such that an independent researcher can reproduce the results" (**?**). On the other hand, Association for Computing Machinery (**?**) divides the problem into three tasks as follows: * **Repeatability** (Same team, same experimental setup): The measurement can be obtained with stated precision by the same team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same location on multiple trials. For computational experiments, this means that a researcher can reliably repeat her own computation.

- **Replicability** (Different team, same experimental setup): The measurement can be obtained with stated precision by a different team using the same measurement procedure, the same measuring system, under the same operating conditions, in the same or a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using the author's own artifacts.
- **Reproducibility** (Different team, different experimental setup): The

measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same result using artifacts which they develop completely independently.

For the needs of this chapter we will use the Vandewalle's definition and treat papers as fully reproducible only when they meet the conditions listed there.

1.2.3 Related Work

Reproducibility is a hot topic. "Open Science in Software Engineering" (**?**) describes the essence of *open source*, *open data*, *open access* and other *openness*. The article mentions that ability to reproduce work is important for the value of research. *Open Science* has many positive effects: increases access and citation counts, supports cooperation through open repositories. "Reproducibility Guide" (**?**) contains a lot of informations and tips on how to make research easier to reproduce. The guide also contains the list of tools that can make our research more reproducible (for example version control and automation. And the most important for us: it includes the checklist of questions that help verify the ability to reproduce. Edward Raff emphasizes the word *independent* in his article (**?**). *Independent reproducibility* means that we can obtain similar results independently of the author and his code.

These articles highlight various aspects of reproducibility. We want to verify how the authors care about reproducibility, what are their biggest reproduction issues and what type of problems can we encounter reproducing articles.

1.2.4 Methodology

1.2.5 Results

1.2.6 Summary and conclusions

1.3 Aging articles. How time affects reproducibility of scientific papers?

Authors: Paweł Morgen, Piotr Sieńko, Konrad Welkier (Warsaw University of Technology)

1.3.1 Abstract

Reproduction of a code presented in scientific papers tend to be a laborious yet important process since it enables readers better understanding of the tools proposed by the authors. While recreating an article various difficulties are faced what can result in calling the paper irreproducible. Some reasons why such situations occur stem from the year when the article was published (for example usage of no more supported packages). The purpose of the following paper is to prove whether this is a general trend which means answering the question: is the year when the article was published related to the reproducibility of the paper. To do so a package CodeExtractorR was created that enables extracting code from PDF files. By using this tool a significant number of articles could be analyzed and therefore results received enabled us to give an objective answer to the stated question.

1.3.2 Introduction

Every article published in scientific journal is aimed at improving our knowledge in certain field. To prove their theories, authors should provide papers with detailed, working examples and extensive supplementary materials to reproduce results. Unfortunately these conditions are not always fulfilled. In a such case, other researchers are not able to verify and accept solutions presented by the author. Moreover, article is not only useless for the scientific community but also for business recipients.

Over the years, several different definitions of reproducibility have been proposed. According to Robert Gentleman and Duncan Temple Lang (**?**), reproducible research are papers with accompanying software tools that allow the reader to directly reproduce methods that are presented in the research paper. Other authors suggest that scientific paper is reproducible only if text, data and code are made available and allow independent researcher to recreate the results (**?**). Second definition emphasize importance of accessibility to data used in researches, therefore it seems to be more suitable and complete interpretation of reproducibility. In addition, in this article, we used scale based on the spectrum of reproducibility, proposed by Roger D. Peng (**?**). In his work, he also mentioned reproducibility as minimal requirement for assessing the scientific value of paper. In the past few years, computing has become essential part of scientific workflow. Some best practices for writing and publishing reproducible scientific article were presented by V. Stodden (**?**). Furthermore, she made brief overview of existing tools and software that facilitate this task. Similar issue was closely described by Kitzes (**?**). Tools created solely for reproducibility in R were proposed by Ben Marwick (**?**) in package rrtools.

Although many articles focus on software or framework solutions for reproducibility problems, analysis of scientific papers reproducibility in the context of release date has, to the best of the authors' knowledge, not been described before. The intention of such research is to find correlations between age of article and its reproducibility. Authors believe that finding these dependencies will allow to calculate estimated life span of data science article. Furthermore, as replicability helps with applying proposed methods and tools, its approximated level might be helpful in estimating usefulness of every scientific article.

1.3.3 Methodology

1.3.4 Results

1.3.5 Summary and conclusions

1.4 Ways to reproduce articles in terms of release date and magazine

Authors: Mikołaj Malec, Maciej Paczóski, Bartosz Rożek

1.4.1 Abstract

1.4.2 Introduction and Motivation

Reproducibility is a topic which is quite diminished in today's science world. Scientific articles should be current as long as possible. Their results should be achievable by reader and be the same. Thanks to that science and business world can take advantage of them.The more article is difficult to reproduce, the chance of using knowledge coming from it is smaller. Many researchers tried to define or give principles for reproducibility. There is article published in 2016: "What does research reproducibility mean?" (**?**) which tried to warn about reproducibility crisis. Article in 2017: "Computational reproducibility in archaeological research: basic principles and a case study of their implementation" (**?**), compered computational reproducibility to archaeological research and give guidelines for researches to use reproducibility in computing research. But these are just two of many articles about reproducibility. Some articles are about tools and techniques for computational reproducibility (**?**). They encourage researchers to compute data using environments like Jupiter (**?**) or R markdown (**?**). Thanks to that readers can reproduce finding on their own. What's new about our approach to the subject of reproducibility is focusing on how can release date and magazine affect the amount of work needed to fully reproduce code or is it even possible. A comprehensive comparison of scientific magazines in terms of reproducibility is yet to be created and this article is our best effort to make it happen. Mikołaj Malec

1.4.3 Related Work

1.4.4 Methodology

1.4.5 Results

1.4.6 Summary and conclusions

1.5 Reproducibility of outdated articles about up-to-date R packages

Authors: Zuzanna Mróz, Aleksander Podsiad, Michał Wdowski (Warsaw University of Technology)

1.5.1 Abstract

1.5.2 Introduction and Motivation

The problem of the inability to reproduce the results of research presented in a scientific article may result from a number of reasons - at each stage of design, implementation, analysis and description of research results we must remember the problem of reproducibility - without sufficient attention paid to it, there is no chance to ensure the possibility of reproducing the results obtained by one team at a later time and by other people who often do not have full knowledge of the scope presented in the article. Reproducibility is a problem in both business and science. Science, because it allows credibility of research results (**?**). Business, because we care about the correct operation of technology in any environment (**?**). As cited from "What does research reproducibility mean?" (**?**); "Although the importance of multiple studies corroborating a given result is acknowledged in virtually all of the sciences, the modern use of "reproducible research" was originally applied not to corroboration, but to transparency, with application in the computational sciences. Computer scientist Jon Claerbout coined the term and associated it with a software platform and set of procedures that permit the reader of a paper to see the entire processing trail from the raw data and code to figures and tables. This concept has been carried forward into many data-intensive domains, including epidemiology, computational biology, economics, and clinical trials. According to a U.S. National Science Foundation (NSF) subcommittee on replicability in science, "reproducibility refers to the ability of a researcher to duplicate the results of a prior study using the same materials as were used by the original investigator. That is, a second researcher might use the same raw data to build the same analysis files and implement the same statistical analysis in an attempt to yield the same results.... Reproducibility is a minimum necessary condition for a finding to be believable and informative." Other notable articles about reproducibility include "Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System" (**?**), "Reproducible Research in Computational Science" (**?**) and "A statistical definition for reproducibility and replicability" (**?**). "Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System" focuses on the variability and reproducibility of the outcome of complete software development projects that were carried out by professional developers. "Reproducible Research in Computational Science" is about limitations in our ability to evaluate published findings and how reproducibility has the potential to serve as a minimum standard for judging scientific claims when full independent replication of a study is not possible. "A statistical definition for reproducibility and replicability" provides formal and informal definitions of scientific studies, reproducibility, and replicability that can be used to clarify discussions around these concepts in the scientific and popular press. In our article we focus on the reproduction of old scientific articles on R and packages, which are still being developed. We want to explore how the passage of time affects the ability to reproduce results using the currently available updated tools. We are therefore testing backward compatibility for different packages and checking what affects the reproducibility of the code. We were unable to find scientific articles on this exact issue. There are articles that give ways to measure reproducibility, as well as articles about packages that help with reproduction. But there are yet no articles that summarize the set of packages in terms of their reproducibility.

1.5.3 Related Work

1.5.4 Methodology

1.5.5 Results

1.5.6 Summary and conclusions

1.6 Correlation between reproducibility of components of research papers and their purpose

Authors: Przemysław Chojecki, Kacper Staroń, Jakub Szypuła (Warsaw University of Technology)

1.6.1 Abstract

1.6.2 Introduction and Motivation

It is common knowledge that reproducibility is a way for science to evolve. It is the heart of the scientific method to revisit pre-existing measurements and to try to reproduce its results. However, the term „reproducibility" itself, as well it is crucial to the scientific methodology, it can be also universal at the expense of unambiguousness and usability.
For the purpose of this paper we will have recourse to the definition introduced by ACM:
Reproducibility - The measurement can be obtained with stated precision by a different team, a different measuring system, in a different location on multiple trials. For computational experiments, this means that an independent group can obtain the same resultusing artefacts which they develop completely independently.
This particular definition ilustrates perfectly how in the course of establishing the meaning of term „Reproducibility", the level of importance of auxiliary measurements and settings of the experiment to the overall results is omitted. It is notably significant misconception especially in the experiments from the field of computional science, when reproducing or even maintaining precise operating conditions is usually impossible.
In the following chapters we will attempt to perform an analysis of reproducibility of the papers submitted to the RJournal, regarding especially presumed objectives of enclosed auxilliary computations and artifacts (i. e. code chunks) in overarching structure of a given paper.

1.6.3 Related Work

Although there are many research papers related to this article, the following three could be perceived as a "basis" for our study.
1. Daniel Mendez, Daniel Graziotin, Stefan Wagner, and Heidi Seibold, 2019[1] provides a definition of reproducibility this article uses, and distinguishes it from replicability.
2. Steven N. Goodman*, Daniele Fanelli and John P. A. Ioannidis, 2016[2] defines multiple interpretations of reproducibility. It further divides and classifies reproducibility, and provides a basis on how one can do it.
3. Victoria Stodden, Jennifer Seiler, and Zhaokun Mab, 2018[3] is an example of how one conducts a study on reproducibility within articles of a specific journal. It also provides a frame of reference to compare results.

In their search the authors have not encountered other research papers that study the aspect of reproducibility this article focuses on. If said papers do not actually exist, then this article could provide insights on previously unexamined aspects of reproducibility.

1.6.4 Methodology

1.6.5 Results

1.6.6 Summary and conclusions

1.7 How active development affects reproducibility

Authors: Ngoc Anh Nguyen, Piotr Piątyszek, Marcin Łukaszyk (Warsaw University of Technology)

1.7.1 Abstract

1.7.2 Introduction and Motivation

The key quality in measuring the outcome of researches and experiments is whether results in a paper can be attained by a different research team, using

[1]https://arxiv.org/pdf/1904.06499.pdf
[2]https://stm.sciencemag.org/content/8/341/341ps12
[3]https://www.pnas.org/content/pnas/115/11/2584.full.pdf

the same methods. Results presented in scientific articles may sometimes seem revolutionary, but there is very little use if it was just a single case impossible to reproduce. The closeness of agreement among repeated measurements of a variable made under the same operating conditions by different people, or over a period of time is what researches must bear in mind. Professor Roger D. Peng, leading author of the commentary and an advocate for making research reproducible by others, insists reproducibility should be a minimal standard (**?**).

There have been several reproducibility definitions proposed during the last decades. Gentleman and Temple Lang (**?**) suggest that by reproducible research, we mean research papers with accompanying software tools that allow the reader to directly reproduce the results and employ the computational methods that are presented in the research paper. The second definition is according to Vandewalle et al. (**?**), research work is called reproducible if all information relevant to the work, including, but not limited to, text, data, and code, is made available, such that an independent researcher can reproduce the results. As said by LeVeque (**?**) the idea of 'reproducible research' in scientific computing is to archive and make publicly available all the codes used to create a paper's figures or tables, preferably in such a manner that readers can download the codes and run them to reproduce the results. All definitions converge into one consistent postulate - the data and code should be made available for others to view and use. The availability of all information related to research paper gives other investigators the opportunity to verify previously published findings, conduct alternative analyses of the same data, eliminate uninformed criticisms and most importantly - expedite the exchange of information among scientists.

Reproducibility has great importance not only in the academic world but also it also plays a significant role in the business. The concept of technological dept is often used to describe the implied cost of additional rework caused by choosing an easy solution now instead of using a better approach that would take longer in software development.

There are papers about using version control systems to provide reproducible results (**?**). The authors presented how we can manage to maintain our goal of reproducibility using Git and Org-Mode. Other researchers have created a software package that is designed to create reproducible data analysis (**?**). They have created a package that contains computational modules, data processing scripts, and research papers. The package is build using the Unix principle to write programs that are simple and do well one thing. The program breaks big data analysis chains into small steps to ensure that everything is going in the right way. Some papers suggest using Docker to make sure our research can be reproduced (**?**).

The main goal of our work is to measure the impact of the active development of packages on the reproducibility of scientific papers. Multiple authors (**??**)

suggest using the version control system as a key feature in creating reproducible research. The second paper also provides evidence, that this is widely known. Git and GitHub were used in over 80% of cases. However, there are two kinds of using a version control system. An author can push software into the repository, to make it easily accessible and does not update it anymore. The second option is to keep the repository up-to-date and resolve users' issues. We have not found any research on how these two approaches impact reproducibility.

1.7.3 Related Work

1.7.4 Methodology

1.7.5 Results

1.7.6 Summary and conclusions

1.8 Reproducibility differences of articles published in various journals and using R or Python language

Authors: Bartłomiej Eljasiak, Konrad Komisarczyk, Mariusz Słapek (Warsaw University of Technology)

1.8.1 Abstract

1.8.2 Introduction and Motivation

Due to the growing number of research publications and open-source solutions, the importance of repeatability and reproducibility is increasing. Although reproducibility is a cornerstone of science, a large amount of published research results cannot be reproduced (**?**). Repeatability and reproducibility are closely related to science.

"Reproducibility of a method/test can be defined as the closeness of the agreement between independent results obtained with the same method on the identical subject(s) (or object, or test material) but under different conditions (different observers, laboratories etc.). (...) On the other hand, repeatability denotes the closeness of the agreement between independent results obtained with the same method on the identical subject(s) (or object or test material), under the same conditions."(**?**)

Reproducibility is crucial since it is what an researcher can guarantee about a research. This not only ensures that the results are correct, but rather ensures transparency and gives scientists confidence in understanding exactly what was done (**?**). It allows science to progress by building on previous work. What is more, it is necessary to prevent scientific misconduct. The increasing number of cases is causing a crisis of confidence in science (**?**).

In psychology the problem has already been addressed. From 2011 to 2015 over two hundred scientists cooperated to reproduce results of one hundred psychological studies (**?**). In computer science (and data science) scientists notice the need for creating tools and guidelines, which help to guarantee reproducibility of solutions (**?**, **?**). There exist already developed solutions which are tested to be applied (**?**).

Reproducibility can focus on different aspects of the publication, including code, results of analysis and data collection methods. This work will focus mainly on the code - results produced by evaluation of different functions and chunks of code from analysed publications.

In this paper we want to compare journals on the reproducibility of their articles. Moreover, we will present the reproducibility differences between R and Python - two of the most popular programming languages in data science publications.There is discussion between proponents of these two languages, which one is more convenient to use in data science. Different journals also compete between each other. There are already many metrics devised to assess which journal is better regarding this metric (**?**). There are no publications related to the reproducibility topic which compare different journals and languages. Although there are some exploring reproducibility within one specific journal (**?**). What is more, journals notice the importance of this subject (**?**). Also according to scientists journals should take some responsibility for this subject (**?**).

1.8.3 Related Work

1.8.4 Methodology

1.8.5 Results

1.8.6 Summary and conclusions

2
Imputation

Imputation

3 Interpretability

Interpretability

3.1 Building an explainable model for ordinal classification. Meeting black box model performance levels.

Authors: Karol Saputa, Małgorzata Wachulec, Aleksandra Wichrowska (Warsaw University of Technology)

3.1.1 Abstract

3.1.2 Introduction and Motivation

In the classification problems, the main goal is to map inputs to a categorical target variable. Most machine learning algorithms assume that the class attribute is unordered. However, there are many problems where target variable is ranked, for example while predicting movie ratings. When applying standard methods to such problems, we lose some informaton, which could improve our model performance.

This paper presents various methods to make an explainable machine learning model for ordinal classification problem. The aim is to achieve better results than 'black box' model does. We will test some existing approaches to ordinal classification and take advantage of analysis and imputation of missing data, feature transformation and selection, as well as knowledge from exploratory data analysis.

Our experiments are based on 'eucalyptus' dataset from OpenML. The dataset's objective is to find the best seedlot for soil conservation in seasonally dry hill country. Predictions are made depending on features such as height, diameter and survival of the plants. Target variable is ordered - it is represented by values 'low', 'average', 'good' and 'best'.

3.1.3 Related Work

3.1.3.1 An approach to ordinal classification

3.1.3.2 Comparing black box to linear model

3.1.4 Methodology

3.1.4.1 Initial preprocessing

The aim of this article was to build the best interpretable model for an ordinal classification problem and comparing it to a black box model. First undertaken step was dividing the data into training and test sets, consisting of 70% and 30% of all data, respectively. Since the considered dataset has a categorical target variable, the division was done using random sampling within each class of the target variable in an attempt to balance the class distributions within the splits. A seed was used to assure the same split in each tested model.

In order to get a legitimate comparison, the data was initially preprocessed in such a way as to assure that both models' performances are compared on the same test set. This initial preprocessing included:

1. deleting the observations with 'none' value in the target variable from both the training set and the test set;
2. deleting observations with missing values from test set, resulting in a 6% decease of the test set observations.

It is important to note that missing values are still present in the training set.

The reason why the missing data is deleted from the test set is that many of the explainable models cannot be run with missing data present. This means that the missing values will have to either be deleted or imputed later on. This leads to a possibility that the explainable model will impute missing data differently than the black box model, resulting in two different tests sets. And, if - instead of imputing missing data - we decide to delete it in order to make running explainable model possible, then the obtained test sets will differ in number of rows, making it impossible to draw any meaningful conclusions. Hence the missing data were deleted from the test set.

3.1.4.2 Running the black box model

The black box model chosen for comparison is an extreme gradient boosting model. After the initial preprocessing the xgboost model was trained on the training set and used to predict results on the test set. As this model can only deal with numerical data, categorical (factor) variables were transformed using one hot encoding. The training proces and prediction were done using the mlr package in R, and the exact model specifications were the following:

- Learner classif.xgboost from package xgboost
- Type: classif
- Name: eXtreme Gradient Boosting; Short name: xgboost
- Class: classif.xgboost
- Properties: twoclass,multiclass,numerics,prob,weights,missings,featimp
- Predict-Type: response
- Hyperparameters: nrounds=200,verbose=0,objective=multi:softmax

The quality of prediction was measured using the AUC (area under ROC curve) measure. This provides the base for this research, to which other models' results will be compared to.

3.1.4.3 Running the basic version the explainable model

We have chosen a tree model to be the considered explainable model, its exact specifications were the following:

- Learner classif.rpart from package rpart
- Type: classif
- Name: Decision Tree; Short name: rpart
- Class: classif.rpart
- Properties: twoclass,multiclass,missings,numerics,factors,ordered,prob,weights,featimp
- Predict-Type: response
- Hyperparameters: xval=0

As this model cannot be run with missing data, they were deleted from the training set before training the model. Another step was deleting one from each of the one-hot-encoded variables (the default function transforms variable with n factor levels into n columns, but n-1 columns are sufficient as the n-th column is a linear combination of the remaining n-1 columns). This model performed worse than the black box model - the outcomes are presented in the Results section of this article.

3.1.4.4 Improving the explainable model

As mentioned before, the explainable model was enhanced by applying existing approaches to ordinal classification, feature transformation and selection and missing data imputation. The refinement process consisted of, but was not limited to, the following:

1. Transforming Latitude variable from factor to numeric.

2. Splitting a multiclass classification problem into 3 binary classification problems, like explained in the An approach to ordinal classification section of this article, and using the rpart model on each of the binary problems.
3. Selecting variables: deleting the site names and specific location tags.
4. Imputing missing data in the training set.
5. Tunning the model.

The third step has a scientific justification. The experiment for which the data was collected was focused on finding the best seedlot for soil conservation in seasonally dry hill country. All the data in this dataset comes from New Zealand, but there is a chance that the results of such experiment would be used for other geographical regions. So far our model was making the prediction based also on specific flat map coordinates and site names, that are present both in the training and the test set. This means it would be impossible to use this model for judging seedlots of eucalypti planted outside of New Zealand. To make this possible, we have decided to take away all the variables that give away the exact position of the seedlots, leaving features such as latitude and the characteristics of plants and their habitat.

After each improvement the model was retrained and the results obtained on the test set were saved and compared with the previous version of the model. If the new change has improved the model's performance on the test set then it became the base for further development. Instead, if it has not improved the model's performance, the previous version of the model was being further developed.

3.1.5 Model explanantion

3.1.6 Results

3.1.7 Summary and conclusions

3.2 Predicting code defects using interpretable static measures.

Authors: Wojciech Bogucki, Tomasz Makowski, Dominik Rafacz (Warsaw University of Technology)

3.2.1 Abstract

3.2.2 Introduction and Motivation

Since the very beginning of the computer revolution there have been attempts to increase efficiency in determining possible defects and failures in the code. An effective method to do so could bring many potential benefits by identifying such sites as early as at the code development stage and eliminating costly errors at the deployment stage. McCabe and Halstead proposed a set of measures that are based on static properties of the code (including basic values, e.g. number of lines of code or number of unique operators, as well as transformations of them, (**?**, (**?**))). In their hypotheses, they argue that these measures can significantly help to build models that predict the sensitive spots in program modules. However, it can be argued that the measures they propose are artificial, non-intuitive, and above all, not necessarily authoritative, not taking into account many aspects of the written code and program (**?**).

To support their hypotheses with, McCabe and Halstead collected information about the code used in NASA using scrapers and then used machine learning algorithms. In this article we use the above data sets to build a model that best predicts the vulnerability of the code to errors. We check whether static code measures (being transformations of basic predictors) significantly improve prediction results for the so-called white box models (e.g. trees and linear regression). Our goal is to build, using simple data transformations and easily explainable methods, a so-called white box machine learning model (e.g. tree or logistic regression) that will achieve results comparable to the black box model (such as neural networks or gradient boosting machines) used on data without advanced measures. We also want to compare the effectiveness of the measures proposed by McCabe and Halstead and compare them with the measures we have generated.

3.2.3 Dataset

Our dataset comes from the original research of Halstead and McCabe. We obtained it by combining the sets from OpenML (**?**) and supplementing them with data from the PROMISE repository (**?**).

It contains data collected from NASA systems written in *C* and *C++* languages. The data is in the form of a data frame containing more than 15000 records. Each record describes one “program module”. – with this generic term, the authors defined the simplest unit of functionality (in this case, these are functions). Each record is described with a set of predictors, which can be divided into several groups:

- Basic measures (such as number of lines of code, number of operands, etc.).

- McCabe's measures how complex the code is in terms of control flow and cross-references .
- Halstead's measures for general code readability.
- Target column (1 if module contains defects, 0 if not).
- Source column we added, specifying from which subsystem the module came (the original 5 datasets came from different systems).

In order to verify our hypotheses, we decided at the beginning to remove the Halstead's measures (which were transformations of the basic measures) from the collection to see if we are able to build an effective black box model without them. We also wanted to remove McCabe's measurements, but the basic measurements that he used to calculate his measurements are not preserved in the collection, so we decided to keep them.

There are not many records with openly missing data in the set ($< 1\%$), however, the values of some columns raise doubts – in the column containing information about the number of lines of code of a given module in many cases there is a value 0, which is not reliable. We consider 0 in this column to be missing data and remove records that have 0 here (almost 2000 records).

3.2.4 Methodology

Our research consist of the following stages:

1. Data exploration.
2. Initial data preparation.
3. Building of black-box and white-box models and comparing them against the relevant measurements.
4. Repeating the cycle:
   (a) Improvement of white box models by modifying their parameters or data.

   (b) Measuring the effectiveness of the models built.
   (c) Analysis of the resulting models.
   (d) Keeping or rejecting changes for further work.
5. Selection of the best white box model and final comparison with the black box model.

We use R programming language and popular machine learning project management packages - mlr and drake (**?**, **?**).

3.2.4.1 Data exploration

At this stage, we are taking a closer look at what the data looks like and we are analyzing their distributions, gaps, correlations and simple relationships.

3.2.4.2 Initial data preparation

This stage consists mainly of merging the data sets, as mentioned earlier, and adding a source column (in fact, we added five indicator columns, which contain one-hot-encoded value, as models generally do not cope well with character columns). Since there were very few data gaps, we are imputing them with the median, because this method is effective and fast.

Imputation is necessary from the very beginning, as many models cannot cope with missing values. Since there were very few missing values, it does not affect significantly the result of those models that models that would still work. We are not carrying out further transformations at this stage because we do not want to disturb the results of the next stage.

3.2.4.3 Starting models

We're building models on this almost unaltered data. We are using one poorly interpretable model (black-box) – random forest, specifically ranger package (**?**), because it is fast and low-effort. Among well interpretable models (white-boxes) used in our work there are:

- logistic regression,
- decision tree,
- k-nearest neighbors algorithm.

We train the models into data that we have divided into five folds with a similar distribution of the decision variable, on which we will perform cross-validation. Then we compare the results using commonly used measure – AUC (Area Under Curve) (**?**), which not only assesses whether the observations are well classified, but also takes into account the likelihood of belonging to a class. We use AUC as the main comparative criterion of the models also in the further part of our work.

3.2.4.4 Improving white-boxes

This is a key part of our work. In the iterative cycle we use different methods to improve the quality of the white box models. After applying each of these methods, we check whether it has improved our score and possibly analyze the model, using statistical methods (residuals analysis) and explanatory machine learning (DALEX package (?)), to draw indications of what should be done next.

We are trying the following methods:

- **Tuning hyperparameters of models** – Default hyperparameters for models are generally good, but in specific cases using specific hyperparameters may yield in better results, so we use model-based optimization for tuning those parameters (?).
- **Logarithmic and exponential transformations of individual variables** – So that linear relationships can be better captured and to reduce the influence of outliers, we transform variables using exponential and polynomial functions.
- **Discretization of continuous features** – Some variables do not have a linear effect on the response variable, even if they are transformed by simple functions like exponential function, sometimes there are clear thresholds – so we can replace the variable with indexes of individual segments. The SAFE algorithm helps with this (?).
- **Generating new columns as functions of other columns** – There may be interactions between variables that cannot be captured by linear models. In order to take them into account, we generate new columns, applying to the rest of them various transformations – we take their inverses, products, quotients, elevations to power, and so on. As a result of these operations, a lot of new predictors are created, which we later evaluate. We also analyze their interpretability, i.e. to what extent they are translatable into an intuitive understanding of such a measure. At this point we also consider Halstead and McCabe's measurements.
- **Oversampling and undersampling** – On the basis of the data set, we generate more observations from the minority class using the SMOTE algorithm (?) and remove some of the observations from the majority class so that the model more emphasizes the differences in characteristics of individual classes.
- **Reducing outliers** – For each variable a two-value vector that indicates the thresholds for which the values are considered as outliers is generated. Then all outliers are changed to the nearest value of obtained earlier vector.

Our goal is to beat black-box model. In our case we chose random forest model from package ranger. As a white-box model we used logistic regression. Results were tested on dataset with different transformations.

3.2.4.5 Selecting the best model

At the end of the process, we select the model that has the highest AUC score for crossvlidation on our dataset.

TABLE 3.1: White-box and black-box comparison

Outliers reduction	Discretization	Logarithm	Ranger	Logistic regression
no	no	no	0.7907	0.7354
yes	no	no	0.7897	0.7431
yes	no	yes	0.7782	0.7435
no	yes	no	0.7593	0.7305

3.2.5 Results

3.2.6 Summary and conclusions

3.3 Using interpretable Machine Learning models in the Higgs boson detection.

Authors: Mateusz Bakala, Michal Pastuszka, Karol Pysiak (Warsaw University of Technology)

3.3.1 Abstract

3.3.2 Introduction and Motivation

3.3.3 Related Work

3.3.4 Methodology

3.3.5 Results

3.3.6 Summary and conclusions

3.4 Optimizing features' transformations for linear regression

Authors: Łukasz Brzozowski, Wojciech Kretowicz, Kacper Siemaszko (Warsaw University of Technology)

3.4.1 Abstract

3.4.2 Introduction and Motivation

Linear regression is one of the simplest and the easiest to interpret of the predictive models. While it has already been thorougly analysed over the years (ref), there remain some unsolved questions. One such question is how to transform the data features in order to maximize the model's effectiveness in predicting the new data. An example of a known and widely used approach is the Box-Cox transformation of the target variable, which allows one to improve the model's performance with minimal increase in computational complexity. However, choice of the predictive features' transformations is often left to intuition and trial-and-error approach. In the article, we wish to compare various methods of features' transformations and compare the resulting models' performances while also comparing their computational complexities and differences in feature importance. Many black box regression models use various kinds of feature engineering during the training process. Unfortunately, even though the models perform better than the interpretable ones, they don't provide information about the transformations used and non-linear dependencies between variables and the target. The goal we want to achieve is extracting features and non-linearities with understandable transformations of the training dataset. To measure the improvement of used methods, we'll compare their performance metrics with black box models' as a ground truth. This will allow us to effectively measure which method brought the simple linear model closer to the black box. Also, we'll take under consideration the improvement of black box model performance. Thanks to this, our article will not only present the methods for creating highly performant interpretable models, but also improvement of the results of black box model.

//cytowania

3.4.3 Related Work

There exist many papers related to feature engineering. We will shortly present few of them.

One of these papers is “Enhancing Regression Models for Complex Systems Using Evolutionary Techniques for Feature Engineering” Patricia Arroba, José L. Risco-Martín, Marina Zapater, José M. Moya & José L. Ayala. This paper describes, how feature transformations in linear regression can be chosen based on the genetic algorithms.

Another one is “Automatic feature engineering for regression models with machine learning: An evolutionary computation and statistics hybrid” Vinícius Veloso de Melo, Wolfgang Banzhaf. Similarly to the previous one, this paper tries to automate feature engineering using evolutionar computation to make a hybrid model - final model is simple linear regression while its features are found by more complex algorithm.

//cytowania

3.4.4 Methodology

The main goal of our research is to compare various methods of transforming the data in order to improve the linear regression's performance. While we do not aim to derive an explicitly best solution to the problem, we wish to compare some known approaches and propose new ones, simultaneously verifying legitimacy of their usage. The second goal of the research is to compare the achieved models' performances with black box models to generally compare their effectiveness, having in mind that the linear regression remains highly interpretable. We also wish to compare the models feature importance to check for notable differences.

The main methods of feature transformation compared in the article include:

1. By-hand-transformations - we will use our expertise to derive possibly the best transformations of the dataset, but in this scenario we do not automate the process. Based on various visualizations, including, but not limited to residual plots, Feature Importance plots, Partial Dependency plots and Accumulated Dependency plots, we aim to maximize the model's performance by hand.
2. Brute Force method - this method of data transformation generates huge amount of additional features being transformations of the existing features. They include e.g. taking square of the variable value or multiplying two variables. While the method is known to provide good results, its complexity is much higher than other methods' and may lead to overfitting.
3. Bayesian Optimization method - we wish to treat the task of finding optimal data transformation as an optimization problem. Once we restrict the transformations e.g. by choosing maximal degree of a polynomial transformation of each variable, we may then search the possible models' space with the use of Bayesian Optimization in order to maximize their performance. This way we may also restrain the model from generating a lrage number of variables, while also hopefully yielding good results.
4. One of our ideas is to use GP (Genetic Programming) to find best feature transformations. We will create a set of simple operations such as adding, multiplying, taking a specific power, for example 2, taking logarithm and so on. Our goal is to minimize mean squared error of linear regression (ridge) after transformations. We will use

one of the variations of the genetic algorithms to create an operation tree minimizing our goal. This method should find much better solutions without extending dimensionality too much or making too complex transformations. We will get linear regression with much better performance without loss of the interpretability. This method will teach linear regression many times, because in each generation each individual requires its own training. However, linear regressions are very fast even for datasets with many rows and many columns, thus computation complexity should not be a problem. Whole conception tries to automate feature enginnering done traditionally by hand. Another advantage is control of model complexity. We can stimulate how the operation trees are made, and reduce or increase complexity at will. Modification of this idea is to add regularization term decreasing survival probability with increasing complexity. At the end model could also make a feature selection in the same way - then one of possible operations in the set would be dropping.

The research is conducted on *Concrete_Data* dataset from the OpenML database. The data describes the amount of ingredients in the samples - cement, blast furnace slag, fly ash, water, coarse aggregate and fine aggregate - in kilograms per cubic meter; it also contains the drying time of the samples in days, referred to as age. The target variable of the dataset is the compressive strength of each sample in megapascals (MPa), therefore rendering the task to be regressive. The dataset contains 1030 instances with no missing values. There are also no symbolic features, as we aim to investigate continuous transformations of the data.

We use standard and verified methods to compare results of the models. As the target variable is continuous, we may calculate Mean Square Error (MSE), Mean Absolute Error (MAE), and R-squared measures for each model, which provides us with proper and measurable way to compare the models' performances. The same measures may be applied to black box models. The feature importance measure used in the after-evaluation comparison is based on the permutation feature importance, easily applied to any predictive machine learning model and therefore not constraining us to choose from a restricted set. In order provide unbiased results, we calculate the measures' values during cross-validation process for each model, using various number of fold to present comparative results.

//cytowania do dopisania

3.4.5 Results

3.4.6 Summary and conclusions

3.5 Surpassing black box model's performance on unbalanced data with an interpretable one.

Authors: Witold Merkel, Adam Rydelek, Michał Stawikowski (Warsaw University of Technology)

3.5.1 Abstract

3.5.2 Introduction and Motivation

Recently, an increase in demand of interpretable models can be seen. Machine learning models have gained in popularity in recent years among many fields of business science, industry and also more and more often in medicine. “Interpretability is a quickly growing field in machine learning, and there have been multiple works examining various aspects of interpretations (sometimes under the heading explainable AI).” (**?**) The problem, however, turned out to be blackbox models, which did not provide sufficient information about the motivation in making specific decisions by the models. ‘’Machine Learning models have been branded as ‘Black Boxes’ by many. This means that though we can get accurate predictions from them, we cannot clearly explain or identify the logic behind these predictions.” (**?**)

Interpretability of models is a desirable feature among specialists in fields other than machine learning, it helps them make better decisions, justify their choices, and combine expert knowledge with the model's indications. ‘’ Machines and humans work differently in how they sense, understand and learn. Machines are better at recognizing low-level patterns in huge amounts of data, while people excel at connecting the dots among high-level patterns. To make better decisions, we need both working together. ‘’ (**?**) Trust and transparency are also demanded.

There are many methods that can help us create an interpretable model ‘’The easiest way to achieve interpretability is to use only a subset of algorithms that create interpretable models. Linear regression, logistic regression and the decision tree are commonly used interpretable models.” (**?**)

Another way may be to use blackboxes to create an interpretable model. They can help us during transformation of the original data set or, for example, in selecting variables. In this article, we will discuss the process of creating an interpretable model whose target effectiveness will be comparable to blackbox models. We will present the whole workflow, during which we will get acquainted with the dataset with which we will work, we will use advanced feature engineering methods and compare the results obtained during all phases of process. An additional problem we will face during work will be unbalanced data and creating a model that will take them into account during prediction.

We will use machine learning tools and frameworks available in R and Python. The subject of our analysis will be the "Adult" data set - (`https://www.openml.org/d/179?fbclid=IwAR3W2OO_QNLM9cLmThzwmtJjOZ-GteprmynSTumIcTyT93BVeTX4gbGbZtM`). The goal is to predict whether income exceeds 50 000 $ annually based on census data.

3.6 Which Neighbours Affected the Price of a House in the '90s?

Authors: Hubert Baniecki, Mateusz Polakowski (Warsaw University of Technology)

3.6.1 Introduction

Real estate value varies over numerous factors. These may be obvious like location or interior design, but also less apparent like the ethnicity and age of neighbours. Therefore, property price estimation is a demanding job that often requires a lot of experience and market knowledge. Is or was, because nowadays, Artificial Intelligence (AI) surpasses humans in this task. Interested parties more often use tools like supervised Machine Learning (ML) models to precisely evaluate the property value and gain a competitive advantage.

The dilemma is in blindly trusting the prediction given by so-called black-box models. These are ML algorithms that take loads of various real estate data as input and return a house price estimation without giving their reasoning. Black-box complex nature is its biggest strength and weakness at the same time. This trait regularly entails high effectiveness but does not allow for interpretation of model outputs. Because of that, specialists interested in supporting their work with automated ML decision-making are more eager to use white-box models like linear regression or decision trees. These do not achieve state-of-the-art performance efficiently, but instead, provide valuable information about the relationships present in data through model interpretation.

For many years houses have been the most popular properties; thus, they are of particular interest for ordinary people. What exact influence had the demographic characteristics of the house neighbourhood on its price in the '90s? Although in the absence of current technology, it has been hard to answer such question years ago, now we can.

In this paper, we perform a case study on the actual US. Census data from 1990 and deliver an interpretable white-box model that estimates the median house price by the region. We present multiple approaches to this problem and choose the best model, which achieves similar performance to complex black-boxes. Finally, using its interpretable nature, we answer various questions that give a new life to this historical data.

3.6.2 Data

Data description. Done but not polished.

3.6.3 Methodology

In this section, we are going to focus on developing the best white-box model predicting house prices median, which provides interpretability of features. Finally, we will compare these results with black-box models.

3.6.4 Results

Better than black-box?? Spoiler alert.

3.6.5 Conclusions

Spoiler: Trees are cool. House prices are ethnically biased.

3.7 Explainable Computer Vision with embeddings and KNN classifier.

Authors: Olaf Werner, Bogdan Jastrzębski (Warsaw University of Technology)

3.7.1 Abstract

3.7.2 Introduction

Computer vision is widely known use case for neural networks. However neural networks are infamous for their complexity and lack of interpretability. On the other hand simple classifiers like KNN have really poor results for complex tasks like image recognition. In this article we will prove that it is possible to get best of both worlds using emmbeddings.

3.7.3 Data

We are going to use dataset Fashion-Mnist[1]. Fashion-MNIST is a dataset of Zalando's article images, consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Classes are following: T-shirt/top, Trouser, Pullover, Dress, Coat, Sandal, Shirt, Sneaker, Bag, Ankle boot.

3.7.4 Methodology

3.7.5 Results

3.7.6 Conclusions

[1] `https://www.openml.org/d/40996`

4

Acknowledgements

This project is inspired by a fantastic book Limitations of Interpretable Machine Learning Methods[1] created at the Department of Statistics, LMU Munich. We used the LIML project as cornerstone for this reopsitory.

[1]https://compstat-lmu.github.io/iml_methods_limitations/