POLITECNICO DI TORINO

master's Degree in Computer engineering

master's Degree Thesis

Fairness and Equality of Opportunity in the Algorithmic Era

Supervisors

Prof. Antonio VETRÒ

Prof. Juan Carlos DE MARTIN

Elena BERETTA

Candidate

Flavio Emanuele CANNAVÒ

APRIL 2020

Summary

In the last few decades, we have been witnessing an increasing widespread diffusion of algorithmic tools among both public and private sector all over the world, especially of automatic decision systems, such as recommendation and ranking systems. Although these kinds of systems have existed for several years, they recently come back on the cutting edge thanks to explosively growing of computational power, data availability, and artificial intelligence algorithms. A number of AI algorithms have recently been developed and employed to enhance forecasting accuracy and to increase the learning capability from users' activity and historical data (which ML methods analyze to extract recurrent patterns and, in the end, knowledge). Many experiments proved that results obtained by ML methods are often more accurate of the ones gathered from years-experienced professional. Furthermore, ML is supposed to be impartial, faster, and capable of uncovering factors which may be relevant but as complex as humans usually overlook them. Besides a long list of advantages, the application of AI&ML systems also leads to a wide range of critical aspects such as data availability and features selection. In fact, the results obtained by ML algorithms are highly sensitive to the data employed in training phase. If those data reflect historical prejudices against certain social groups, prevailing cultural stereotypes, and existing demographic inequalities, without specific intervention, machine learning will encode stereotypes, including wrong and harmful ones, in the same way that it encode useful information. Over the last few years, evidences of inequality and systematic discriminations arising from thoughtless and unaware application of such tools have intensified. In light of the current ubiquity of this technology, the crucial influence on people's careers and business opportunities, educational placement, access to benefits, and even social and reproductive success is commonly agreed. In order to inspire and foster a more careful and responsible development, researchers promoted the debate on the introduction of moral notions in machine learning algorithms in order to make them more compatible with our society. Many ongoing researches show a wide and different set of attempts to formalize the concept of fairness in AI&ML domain. The simultaneous adoption of equity criteria and methods that embed interdisciplinary concepts in algorithmic systems, may not only mitigate undesirable

outcomes such as bias and discrimination, but also produce positive effects and reduce social inequalities. Our research is based on these assumptions. Our main aim is the integrations and exploitation of philosophical, legal and economic sciences into ranking systems area, in order to equity-aware models. Starting from previous research in this ground, our models acts as countermeasure against inequalities and diversity of our society, mitigating them and providing a fairer and less discriminatory outcome. Our ranking systems are based on Roemer's theory of Equality of Opportunity , whose main foundation is based on the assumption that the individual's achievement should depend on choice, effort, and ability, not on the circumstances of birth.

Table of Contents

List of Tables

List of Figures

Acronyms

AI
: artificial intelligence

AI/ML
: artificial intelligence and machine learning systems

ADS
: automated decision systems

AUC
: area under the curve

EOp
: equality of opportunity

ML
: machine learning

RS
: recommendation systems

MLP
: multi-layer perceptron

DD
: demographic disparity

Chapter 1

Introduction

The availability of large-scale data, particularly on human activities, is profoundly changing the world in which we live: universities, companies, governments, financial institutions and non-governmental organizations are actively experimenting and adopting Automated Decision Systems (ADS) - “more commonly known as algorithms” [1] - that aim to aid or replace human decision making by learning from our behavior.

The application of software automated techniques in decision-making processes is extremely tricky, especially when these systems are powered with artificial intelligence which introduces additional concerns and complexity. As proven by many researchers, often these systems are affected by a number of ethical and legal issues related to transparency, accountability [2], bias and discrimination [3], that has led to intended and unintended negative consequences, such as disproportionate adverse outcomes for disadvantaged groups [4]. Recent scandals such as the one involving Cambridge Analytica and Facebook [5] or the study conducted by ProPublica on the COMPAS Recidivism Algorithm [6] are two illustrative examples of the relevance of the issues for our society. With the worldwide growing adoption of this technology, concerns appeared also in Europe spreading in different areas such as healthcare, housing, policing, education, justice, and job-placement. Many of the tools adopted to take decisions, evaluating scores, doing predictions, end up in discriminatory behavior and often it is not possible to shed some light on them because considered trade secret.

To try to overcome these problems, or at least to mitigate them, experts begun to study in depth the effects of such systems on people’s life not only from a technical point of view. Many researchers are fostering the debate on the introduction of moral notions in machine learning algorithms in order to make them more compatible with our society. The adoption of methods from the philosophical, legal and economic sciences into computational systems aims not only to avoid undesirable outcomes such as bias and discrimination, but to produce positive effects and reduce social inequalities. Most popular solutions of ongoing researches show a wide and different set of attempts of formalize the concept of fairness. Such solutions inspired and regulated a more careful and responsible development and diffusion of artificial intelligence and machine learning systems (AI/ML).

The debate about the introduction of ethics into AI/ML is still open and constantly evolving. A universally valid solution is far from be reached. Due to their multidisciplinary nature, the approaches proposed are limited to specific context of use, remaining strictly dependent from a given political, economic and social environment. In our work we try to adopt a method based on theory of social justice and reallocation of resources. We show a comparison of the outcome of different ranking system based on 3 different dimensions of Equality of Opportunity, in the context of a students' selection process. We examine the benefits for the global population given by our approach in terms of fairness and reduction of social inequality.

Chapter 2

Background

Big Data are shaping our life. Every second billions of data flow from one side of the world to the other under the will - and the rules - of hundred of thousands of processes. We are used to say our world is a data-driven world but, as a matter of fact, data are just data, and they are part of bigger systems who decide when they are produced, how, and where. These systems, belonging to *cyberspace*, today more and more are overlapping our biological space, our everyday life. We know that every system is made up by processes, and cyber-processes are entities governed by some logic, who implements a finite sequence of well-defined, computer-implementable instructions: in other words, algorithms. In the cyberspace, through some algorithm, practically every action of our everyday is recorded, stored, analyzed, aggregated and transformed in the *quantified selves* of ourselves[2]. This is a great opportunity for the algorithms' owners - companies - who can extract some knowledge from our daily routines, when we chat with our friends, when we drive, when we listen to music, when we read the latest news, when we order food, etc. Companies exploit that knowledge in many ways, often with a unique aim: to increase their profit. They use AI-powered algorithms fed with our data in order to make decisions for ourselves, and for give us *suggestions*: the friend we wish to contact, the song we would like to listen, the news we are most interested in. For people who are too lazy, or lost in the frenetic pace of their life, suggestions may turn into indications. In this sense we say algorithms are shaping our life. Things begin to get more serious if we consider the same data and the same predictive algorithms are used for much more delicate aspects of our life. The *knowledge* companies got from our data is used to decide to grant a loan or deny it, if one person will be recidivist in two years or not, if a job seeker could be a good employee or a total waste of resources. Usually these decisions and suggestions are performed by two different kind of systems: automated decision systems and ranking systems. These systems are relatively old but in recent years they come back on the cutting edge thanks to explosively growing of computational power, data availability, and artificial intelligence algorithms.

2.1 Automated Decision Systems

"Algorithmically controlled, automated decision-making or decision support systems (ADS) are procedures in which decisions are initially—partially or completely—delegated to another person or corporate entity, who then in turn use automatically executed decision-making models to perform an action" [7]. These kind of systems are in use all over the EU, and more and more countries are deciding to rely on them contributing to widespreading of the technology in many areas such as healthcare, housing, policing, education, justice, etc. (i.e. in December 2018 the European Commission presented a Coordinated plan including 70 joint actions for closer and more efficient cooperation between Member States to foster the development and use of AI in Europe). ADS' aim is minimizing human intervention in an ongoing decision-making process. Their purpose is to sense situations, employ codified knowledge, and react properly with marginal human involvement. Current systems have roots in both artificial intelligence and decision-support tools, in that they often involve both business-rule processing and statistical or algorithmic analysis [8].

2.2 Recommendation Systems

Differently from ADS Recommendation systems (RS) are not involved directly in decision-making processes but give to users related *suggestions* such as what items to buy, what news to read, what music to listen to [9]. As for ADS and AI also the growth of RS goes hand in hand with the (big)data production, as countermeasure to users overflooding caused by the last years overwhelming availability of information. In fact RS aim to orient users between a multitude of contents and companies adopt them for different reasons: increase the number of items sold, increase items diversity, enhance user satisfaction, increase the user perceived fidelity, better understand user needs. RS may be very different from each other, in fact their design and implementation are strictly dependent on the application's domain. However it is possible to identifying some macro categories based on the kind of knowledge the RS exploits, and its recommendation algorithm (i.e. how the prediction is performed). We distinguish:

- **Content based:** this method is based on the representation of items and their features. The system should be able to estimate similarity between items and to record items liked from the users. Items recommended are similar to the ones that the user liked in the past.

- **Collaborative filtering:** recommendations are based on historic users' preference. The items liked from a user are recommended to other users with similar taste.
- **Demographic:** items liked from a user are recommended to users who share same/similar demographic characteristic.
- **Knowledge-based:** items are recommended dependently on specific domain knowledge about how certain item features meet users needs and preferences.
- **Community-based:** this technique follows the epigram "Tell me who your friends are, and I will tell you who you are" [10]. It implies that a user get recommendations based on its friends preferences.
- **Hybrid RS:** this approach use information from both user-item interactions and users/items' characteristics. The methods above are combined exploiting the advantages and mitigating the limitations.

2.3 Ranking Systems

Ranking algorithms constitute the core of search and recommendation systems for several applications requiring a composition of a sorted list based on certain attributes, such as hiring, lending, and college admissions [11]. The ranking process usually involves computation of the score of each individual from some data set, sorting the individuals in decreasing order of score, and finally returning either the full ranked list, or its sub-set which contains the highest-scoring items, the top-k. The items to be sorted from ranking systems are artistic products, job candidates, or other objects that transfer economic value, and it is widely recognized that the position of an item in the ranking has a crucial influence on its career and business opportunities, educational placement, access to benefits, and even social and reproductive success [12]. It is therefore of societal and ethical importance to investigate whether such algorithms provide outcomes that can declass, demerit, or exclude individuals of disadvantaged groups (e.g., racial or gender discrimination) or promote products with displeasing characteristics (e.g., gendered books)[13].

Chapter 3

The Discrimination Problem

In this chapter we select and analyze fourteen real cases of algorithmic discrimination resulting from the incorrect design of automatic systems and/or decision tools. These systems used in conjunction with ML algorithms are really sensitive to the data used in the training phase, i.e. a development phase in which the model learns to recognize a common pattern in the data provided. In the following section we will provide technical reasons which may explain the causes of these undesired behaviors. The examples provided are extracted from well known newspapers and from papers released to the scientific community. Particular attention is paid to European cases, although they represent a minority of evidence in literature.

3.1 Why Machine Learning May Lead to Unfairness: Evidence from Risk Assessment for Juvenile Justice in Catalonia [14]

State: Spain
Year of publication: 2019
Domain: Justice
Discrimination problem: The ML models marks as recidivist male defendants, foreigners, or people of specific national groups more frequently than others

Researchers propose a methodology to assess predictive performance and unfairness of Machine Learning algorithms used in juvenile recidivism prediction. Results are compared to SAVRY, a common existing risk assessment tool. These kinds of tools are globally adopted to support the judges by providing them the defendant's risk of recidivism, but they are different in the way of getting the results: retaining a certain degree of freedom and human involving effort or following a more inflexible procedure. Such instruments are called Structured Professional Judgement (SPJ) and there are a couple of reasons for preferring them. In fact, judges are naturally not free from subjective biases, and meta-studies proved that structured assessment may predict criminal behaviour better than individual experts. Furthermore, structured assessment usually is less severe when providing punishments. In particular, the Structured Assessment of Violence Risk in Youth (SAVRY) is used to assess the risk of violence in juvenile justice, and it leaves to professionals a high degree of freedom. It is thought to design intervention planning (i.e. clinical treatment, or release and discharge decision), and plays a crucial role in the course of a juvenile defendant in the justice system. Researchers use a Catalonian dataset of 4753 adolescents offenders, aged 12-17 years, finished a sentence in the juvenile justice systems in 2010, for crimes committed between 2002 and 2010. The research is focused on the sub-sample of 855 offenders who were subject to a SAVRY assessment which predicted recidivism for a period between release in 2010 and December 31, 2015. To conduct experiments, they were made different settings depending on the selected features:

- Static ML. It includes static features such as demographics and criminal history, such as sex, and nationality.
- SAVRY ML. It includes all SAVRY features, the final expert evaluation, the 24 risk items, the corresponding summary scores, the six protective features, the five average scores on individual characteristics as well as the program that the defendant was in (internment or probation).
- Static + SAVRY ML. The conjunction of 1 and 2.

As baselines, it is considered SAVRY Sum, the summed score of all SAVRY risk items, and the “Expert” evaluation. The researchers reported statistical models that achieved the best predictive results in terms of area under the curve (AUC), namely logistic regression and multi-layer perceptron (“mlp”). As fairness evaluation metrics are considered demographic parity and error rate balance.Demographic parity means that each person belonging to a certain group (having a certain protected attribute), has the same probability of being classified as recidivist as someone from the reference group (having another specific protected attribute). In the results it showed a derived metric, the demographic disparity (DD), which is the ratio of the groups probabilities expressed above (i.e. DDi = 2 means that someone with at-tribute ai is twice as likely to be classified as recidivist as someone from the reference group with attribute ar). Error rate balance means that each person belonging to a certain group (having a certain protected attribute), has the same probability of being falsely classified as recidivist (or non-recidivist) as someone from the reference group (having another specific protected attribute). Results show a derived metric, the false positive (or negative) rate disparity (FPRD/ FNRD), which is computed simply dividing the FPR (or FNR) of a certain group for the FPR (or FNR) of the reference group. In order to study predictive performance, it was compared the result of ML methods when using SAVRY features, and without them. The experiment shows that not including demographic and criminal history features decreases the accuracy across all methods with values between (.01, 0.05) points, this means that although informative for an evaluator, the SAVRY features are less useful for ML methods in determining if a person will be recidivist. Furthermore, as expected from data-driven methods, combining features derived from SAVRY items with static demographics and criminal history, or increasing the size of the training set yields better AUC across several learning algorithms. Considering gender from Figure 3.1 we can see that "SAVRY Sum" is within the

Figure 3.1: Comparison of group fairness metrics using sex as the protected attribute. Reference group is men (extracted from the original paper).

fairness bounds in terms of FPRD, while all the other are less likely to erroneously label females as recidivists than men. The ML methods, while staying included in the acceptable range when using SAVRY features, begin to be discriminatory when adopting demo-graphic features, with women being more likely to be classified as non-recidivists. Considering all the three metrics we notice that training on static non-SAVRY features foster the disparity between the two groups, with slight differences depending on the learning algorithm used. The results in terms of nationality displayed in figure 3.2 show that ML methods have higher disparity than the "SAVRY Sum" and the expert evaluation across all metrics. Foreigners are more likely to be falsely labelled as recidivist (FPRD), they are less likely to be labelled as non-recidivists (FNRD) and their proportion of individual labelled as recidivists is higher (DD).

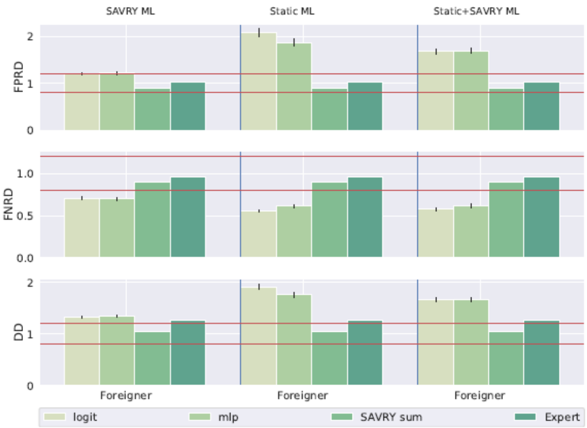

Figure 3.2: Comparison of group fairness metrics in terms of nationality. Reference group is Spanish (extracted from the original paper).

3.2 Austria's employment agency rolls out discriminatory algorithm, sees no problem [15]

State: Austria
Year of publication: 2019
Domain: Employment
Discrimination problem: Among candidates with the same experience and qualifications the algorithm penalizes women

The Austrian employment agency is a state-owned company in charge of helping job seekers. It announced that it would start a collaboration with an external contractor, Synthesis Forschung, in order to develop a system that automatically give a score to each job seeker based on many features, in order to guarantee that the agency does not waste resources on giving help to people who will not gather any benefit from it. Job seekers will be categorized as one of the following: group A are people who need no help in finding a new job, group B are people who might benefit from retraining, and group C are people considered unemployable, who will obtain less help from AMS and may be discharged to other institutions. The algorithm is made of a series of statistical models based on past employment records. The researchers ran statistical regressions to find out which factors were best at predicting an individual's chances of finding a job. The process does not infringe article 22 of the General Data Protection Regulation that prohibits purely automated decision-making on individuals, because AMS case workers retain the possibility of assessing any kind of verdict bypassing the algorithm's judgement.

Studies show that, under a certain model, women are given a negative weight, as are disabled people and people over 30. Women with children are also negatively weighted but, remarkably, men with children are not. In other words, the algorithm's tendency is to put women in a lower group even if her experience and qualifications are comparable with the ones of a man. The AMS algorithm has been widely criticized in Austria but AMS only released 2 of the 96 statistical models claimed to be used to assess job seekers, unsatisfying any demand of transparency.

3.3 Black box Schufa [16]

State: Germany
Year of publication: 2019
Domain: Financial
Discrimination problem: Younger people and males in general receive a higher financial risk estimation

Schufa is the most prominent credit agency in Germany who claims to have information on more than 67 million consumers and provides a score for each of them on which hundreds of banks, telco providers, and retailers rely and exploit to support their business (i.e. it is used to determine which users get to see which ad, or which customers get a loan.). How this score is obtained is a business secret and it is not possible to accurately understand how the Schufa evaluation algorithm works. A big crowdsourcing project involved 2,800 volunteers that asked Schufa for their free personal credit report and shared those documents with the community of investigative reporters, help them to recognize systemic irregularities in the scoring, and skimpy that Schufa knows way less about many people than one commonly think. For the 23.7% of the people in the dataset, Schufa has stored a maximum of three pieces of business information, such as the opening of a current account and the termination of a mobile phone contract and a credit card. Instead it only owns vague information such as addresses, age and gender. And again, more than 20 consumers whom Schufa are certified to have a "satisfactory to increased risk", even though their financial history does not include more than three entries, which are all positives. Schufa calculates a value between 0 and 10,000 points for each person combining their respective stored data in such a way that changes accordingly to the company who requires the score. The key point is that higher score is better, but for each score variant only a limited number of about 15 different repayment probabilities is transmitted. That means small details can lead to a consumer slipping into the next worse category, being constrained to suffer all the related consequences. It is also apparent that information such as date of birth, gender and number of stored addresses play a role in the risk estimation. For instance in the whole dataset, younger people are frequently ranked worse than older ones. Although they have otherwise similar features. Furthermore, some minus sign, which seem correlated to some kind of penalization, are more frequently found in males than in females. The fact that age and gender are included in the score is actually not prohibited, in fact the General Equal Treatment Act which aims to protect consumers from discrimination founded on age and sex is not valid with respect to credit agencies.

3.4 Regulator looking at use of facial recognition at King's Cross site [17]

State: United Kingdom
Year of publication: 2019
Domain: Privacy & Security
Discrimination problem: Black people are investigated more frequently of others because erroneously marked as potentially suspicious

The UK's privacy regulator decided to examine the facial recognition technology used in CCTV systems at the King's Cross development in central London and belonged by property companies, because concerned on its legality, in fact the use of this technology, specially from private companies, should be strictly necessary and compliance with the law. Many people have criticized the facts, assessing that these kind of systems harm people privacy and freedom of expression and there is no transparency about how are being deployed and who they are targeting. It is known that cameras using the software are used by police forces, together with specific smartphone applications, to scan faces in large crowds in public places such as streets, shopping centres, football stadiums and music events such as the Notting Hill carnival, and compare them to a database of suspects (or other persons of interest). Researchers from Essex University were asked by the Met police to study the force's trials of its facial recognition software and concluded that only 19% of the 42 cases examined could they be 100% sure the force had recognized the right person. Of course, also in this case remain valid all the considerations and the concerns that facial recognition technology has a racial bias, that it is less effective in accurately distinguishing black people.

3.5 UK launched passport photo checker it knew would fail with dark skin [18]

State: United Kingdom
Year of publication: 2019
Domain: Privacy & Security
Discrimination problem: Black people are constrained to use old checking-in procedure because the newest system does not work properly

In June 2016 the UK government enhanced with a face-detection system its passport photo checking service, despite knowing the technology had some big issue recognizing people belonging to some ethnic minorities, creating in fact a racialist disparity in experience between users. What is critical in this case, it is not the technology working inappropriately (we already are aware of issues in detecting faces of people with darker shades of skin), but the fact that government decided to deploy the buggy system, ignoring the possible consequences.

3.6 Prisoner risk algorithm could program in racism [19]

State: United Kingdom
Year of publication: 2019
Domain: Justice
Discrimination problem: The algorithm tends to put non-white prisoners in high-security cells more frequently than white ones

It was launched in the UK a new digital tool that exploits data from police, National Crime Agency, and prison service, to categorize prisoners of English jails. Categorization assesses the security restriction necessary for a given person, deciding in fact if a prisoner will be detained in a low secure jail or in a more isolated one. This kind of decision not only affects how strictly the offender will be controlled but also his rehabilitation opportunities. The authors of the article found that the new algorithmic system could be affected by racial bias, and it has a tendency to unfairly classify ethnic minority prisoners as high security requiring. The investigation performed in August 2018 indicates that the new algorithm penalized the 16% of non-white prisoners, signalling more severe requirements. Instead, only the 7% of white prisoners have suffered the same increment in their security category. The Minister of Justice pointed out that the new tool should be used only as a support for the categorization process, and it has not the full decisional power. Furthermore, prisoners should preserve their faculty to appeal the categorization and have access to the justification for all decisions made.

3.7 Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification [20]

Year of publication: 2018
Domain: Gender Classification
Discrimination problem: Despite accuracy of some gender recognition system are claimed to be very high, performances on darker people are significantly poorer than the average

The research is related to the performances of automated facial image analysis, which describes a range of face perception tasks, such as face detection, face classification, and face recognition. In particular, the authors of the research evaluated 3 commercial gender classification systems and showed that they are affected by consistent performance inequality among different classes, race and gender. In order to do that, due to the phenotypic imbalances in already existing benchmarks (displayed in Figure 3.3), the authors introduced a new face dataset composed of 1270 unique individuals, with more balanced gender and skin representation. The proposed dataset is the first one who provides skin's description exploiting the Fitzpatrick six-point skin type scale, a scale that represents the gold standard for skin classification and risk detection used by dermatologist. This dataset, which represents a significant improvement in gender classification benchmarking, is called Pilot Parliaments Benchmark (PPB) and consists of individuals belonging to three different African countries (Rwanda, Senegal, South Africa) and three European countries (Iceland, Finland, Sweden), who was selected for their gender parity in the national parliaments. The research analysed gender classifiers provided by Microsoft, IBM, and Face++. It was observed that face recognition systems work better on local population (with respect to the company who developed them), that is why Face++, which is a Chinese company, was chosen to see if the same observation holds for gender classification. They were assessed the overall classification accuracy, male classification accuracy, and female classification accuracy, plus other metrics, true positive rates (TPR), false positive rate (FPR), error rate, and positive predictive value (PPV). Performance was measured for aggregated group, and for different combinations of subgroups: all subjects, male subjects, female subjects, lighter subjects, darker subjects, darker females, darker males, lighter females, and lighter males. Final results showed that the gender classification performance on female faces are significantly lower than performance on male faces, across all classifiers. The differences between the two error rates range from 8.1%

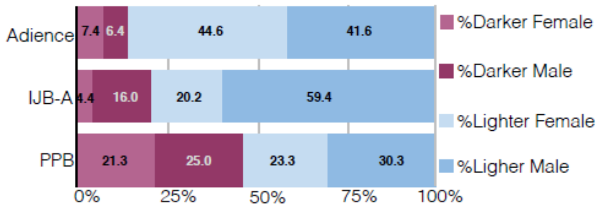

Figure 3.3: Reference corpora distributions compared with PPB dataset distribution (extracted from the original paper)

to 20.6%. Furthermore, all classifiers perform better on lighter faces than darker ones, with differences of error rate that is between 11.8% and 19.2%. The most bias affected subgroup is the one of darker female faces. All classifiers perform worse on it, with 20.8% - 34.7% of error rate.

Set	n	F	M	Darker	Lighter	DF	DM	LF	LM
All Subjects	1270	44.6%	55.4%	46.4%	53.6%	21.3%	25.0%	23.3%	30.3%
Africa	661	43.9%	56.1%	86.2%	13.8%	39.8%	46.4%	4.1%	9.7%
South Africa	437	41.4%	58.6%	79.2%	20.8%	35.2%	43.9%	6.2%	14.6%
Senegal	149	43.0%	57.0%	100.0%	0.0%	43.0%	57.0%	0.0%	0.0%
Rwanda	75	60.0%	40.0%	100.0%	0.0%	60.0%	40.0%	0.0%	0.0%
Europe	609	45.5%	54.5%	3.1%	96.9%	1.3%	1.8%	44.2%	52.7%
Sweden	349	46.7%	53.3%	4.9%	95.1%	2.0%	2.9%	44.7%	50.4%
Finland	197	42.6%	57.4%	1.0%	99.0%	0.5%	0.5%	42.1%	56.9%
Iceland	63	47.6%	52.4%	0.0%	100.0%	0.0%	0.0%	47.6%	52.4%

Figure 3.4: Gender classification performance as measured by the positive predictive value (PPV), error rate (1-PPV), true positive rate (TPR), and false positive rate (FPR) of the 3 evaluated commercial classifiers on the PPB dataset. (Extracted from the original paper).

3.8 The Risk of Racial Bias in Hate Speech Detection [21]

Year of publication: 2019
Domain: Hate Speech Detection
Discrimination problem: Common dialect words used by minority group in normal circumstances are often recognized as offensive

This research is about investigation in several widely used Twitter corpora annotated for toxic content, in order to find potential biases which can be propagated on toxic language detection models trained with them. Toxic language, such as hate speech, abusive speech, or any kind of offensive speech, is a problem which is becoming more and more present on social media platforms, and represents a serious problem that companies need to address, due the potential implication (e.g. real life violence) on society and minority groups which are affected primarily. The task of detecting and removing such content is anything but easy because the risk, especially through automated systems, is to censor or to suppress already marginalized voices. Researchers detected and characterized the racial bias in

	category	count	AAE corr.
DWMW17	hate speech	1,430	−0.057
	offensive	19,190	0.420
	none	4,163	−0.414
	total	**24,783**	
FDCL18	hateful	4,965	0.141
	abusive	27,150	0.355
	spam	14,030	−0.102
	none	53,851	−0.307
	total	**99,996**	

Figure 3.5: Number of tweets in each category, and correlation with AAE (extracted from the original paper).

already annotated corpora for toxic content detection, DWMW17 and FDCL18, establishing strong correlation between toxicity detection and words usually associated with certain minority (all details in Figure 3.5). They used the African American English dialect (AAE) as a proxy for race, which is a widely used dialect of English that is common among those who identify as African American. In addition, they used a specific lexical detector model that yields probabilities of a tweet being AAE or White-aligned English. Specifically, the strongest correlation was with the "offensive" label from DWMW17 ($r = 0{:}42$) and with the "abusive" label from FDCL18 ($r = 0{:}35$). In the end, researchers trained a classifier for each of the two toxic language biased corpora, and tested them on two datasets, DEMOGRAPHIC16 and USERLEVELRACE18. Results displayed in Figure 3.6 show that, while both models achieve high accuracy, the false positive rates (FPR) for the two groups, AAE and White, are very different. The DWMW17 classifier predicts almost 50% of non-toxic AAE tweets as offensive. The FDCL18 classifier presents higher FPR for the “Abusive” and “Hateful” categories for AAE, and higher FPR (of about 5 times) for “None” category for White group. Same tendency, less strong, is present in DWMW17 also. Examining average probability mass of toxicity classes in DEMOGRAPHIC16 and USERLEVELRACE18, it was showed huge disproportion between groups. Specifically, in DEMOGRAPHIC16, AAE tweets are more than twice as likely to be labelled as “offensive” or “abusive”. In USERLEVELRACE18, African American authors tweets are 1.5 times more likely to be labelled as “offensive”.

Within dataset proportions

			% false identification		
DWMW17	Group	Acc.	None	Offensive	Hate
	AAE	94.3	1.1	**46.3**	0.8
	White	87.5	**7.9**	9.0	**3.8**
	Overall	91.4	2.9	17.9	2.3

			% false identification		
FDCL18	Group	Acc.	None	Abusive	Hateful
	AAE	81.4	4.2	**26.0**	**1.7**
	White	82.7	**30.5**	4.5	0.8
	Overall	81.4	20.9	6.6	0.8

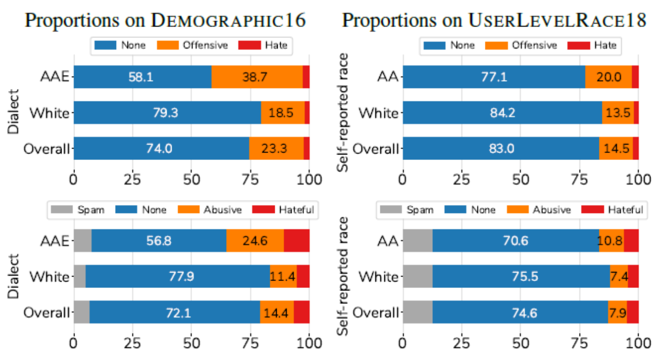

Figure 3.6: Left: classification accuracy and per-class rates of false positives (FP) on test data for models trained on DWMW17 and FDCL18, where the group with highest rate of FP is bolded. Middle and right: average probability mass of toxicity classes in DEMOGRAPHIC16 and USERLEVELRACE18, respectively, as given by classifiers trained on DWMW17 (top) and FDCL18 (bottom). Proportions are shown for AAE, White-aligned English, and overall (all tweets) for DEMOGRAPHIC16, and for self-identified White authors, African American authors (AA), and overall for USERLEVELRACE18.(extracted from the original paper)

3.9 How Amazon’s Algorithms Curated a Dystopic Bookstore [22]

Year of publication: 2019
Domain: Recommendation System
Discrimination problem: The algorithm in charge of deciding and promoting contents on the platform is easily gameable by creators and coordinated groups of users

The author of the article takes into account curation algorithms and their potential consequences on our society. These algorithms are engineered to show us things we are statistically likely to want to see, content that people similar to us (from algorithm's perspective at least) have found appealing, even if that content is objectively unreliable or potentially dangerous, like health-related misinformation. The issue is particularly noticeable in big platform like Amazon, where it influences what millions of people buy, watch, read, and listen to each day. The popular company offers plenty of varieties of recommendation algorithms, like "customers also shopped for" and "customers who bought this item also bought". Furthermore, there are "sponsored" products (which are essentially ads), and finally there's "frequently bought together" a feature that links products across categories. This is considering only the e-commerce section (there are many other such as Amazon Video, Amazon Music, etc.). Customers consider the number of stars and amount of reviews as a proxy for quality. In fact, Amazon tends to reward the high success items putting them on evidence on the first positions of search results and emphasizing them with label as "Amazon's choice" or "Best-selling". Among millions of items is crucial for the sellers to do the best effort to gain visibility for their products, being the item popularity a key input for the algorithm, they learned how to influence the algorithm evaluation in different ways. One of the most popular fraud is to buy or incentivize customers positive reviews by offering discounts or gifts. This is of course a big damage for all the consumers. Communities of true believers exploit their capability of generating high volume traffic in order to increase items popularity. This is exploited to convey and highlight their message, often controversial (e.g. anti-vax movement), on the platform. This is the case of Vaxxed, a movie devoted to the conspiracy theory that vaccines cause autism, whose very high popularity led the algorithm to accidentally promote it for free with a splash page on Amazon's Prime Streaming video platform. In fact, talking about entertainment content, the situation can be even more misleading. Amazon allows content creators to select their own categories and keywords, and it's easy to figure out how this feature can be exploited to let some content pass as something else more popular and more reliable (i.e. pseudoscience books tagged as medicine ones). Amazon is taking incremental steps toward limiting health misinformation, but main efforts arrived primarily only when under significant pressure, and still is not clear how the problems above will be faced.

3.10 Amazon ditched AI recruiting tool that favored men for technical jobs [23]

Year of publication: 2018
Domain: Employment

Discrimination problem: The automatic hiring tool systematically discard women job applications

Since 2014 Amazon's team had been working on an experimental hiring tool used artificial intelligence to give job candidates scores, ranging from one to five stars. The company set up a team in Amazon's Edinburgh engineering hub that grew to around a dozen people, with the goal of developing AI that could rapidly crawl the web and spot candidates worth recruiting. The company developed 500 statistical models dedicated on specific job functions and locations. They trained each model to identify some 50,000 terms that were found on past candidates' curricula. Unfortunately, due to the male dominance across the tech industry in the last 10 years, the data observed to train the model was significantly biased. "The technology favoured candidates who described themselves using verbs more commonly found on male engineers' resumes, such as "executed" and "captured" , instead it penalized résumés that included the word "women's", as in "women's chess club captain" or even downgraded graduates of two all-women's colleges" – we read on the original paper. The result was that automated system built was not gender-neutral when considering candidates for software developer jobs and for other technical posts. Moreover, complications with the data that underpropped the models' decisions meant that unqualified candidates were often recommended for all types of jobs, in fact nullifying the system results who look like almost casual, so that company's recruiters looked at the suggestions produced by the tool when searching for new employee, but never trusted exclusively on those rankings.

3.11 To predict and serve [24]

State: USA
Year of publication: 2016
Domain: Crime Detection
Discrimination problem: The algorithm suggests stronger patrols always in the same location

The paper is about risks associated with the use of police-recorded data on predictive policing systems. In particular it was investigated an algorithm developed by PredPol with drug crime records in Oakland. Predictive policing is the application of analytical techniques to identify future offenders, highlight trends in criminal activity, and even forecast the locations of future crimes. The PredPol algorithm uses a sliding window approach to produce a one-day-ahead prediction of the crime rate across locations in a city, using only the previously recorded crimes. The areas with the highest predicted crime rates are flagged as "hotspots" and receive additional police attention on the following day. The main hypothesis here is that police databases do not constitute a representative random sample of all criminal offences, but for historical and sociocultural reasons tend to over-represent certain minority groups. To investigate the effects that such a biased dataset could have in

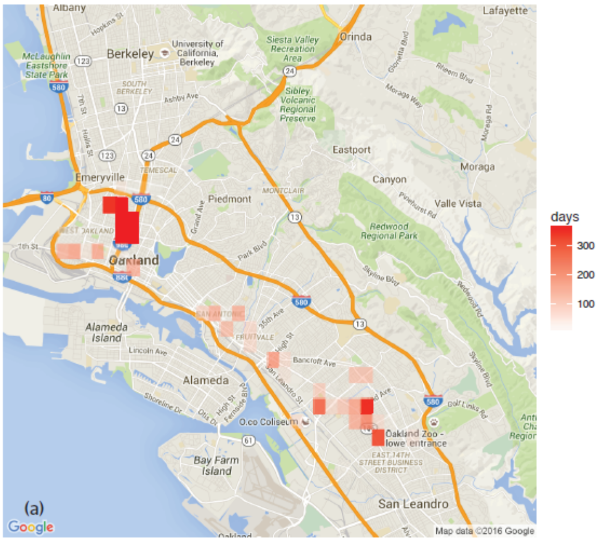

Figure 3.7: Number of days with targeted policing for drug crimes in areas flagged by PredPol analysis of Oakland police data. (extracted from the original paper)

the model, the researchers applied the algorithm to Oakland's police database to obtain a predicted rate of drug crime for every grid square in the city for every day in 2011 and recorded how many times each grid square would have been flagged by PredPol for targeted policing. They found that rather than correcting for the apparent biases in the police data, the model reinforces the existing ones. The locations that are flagged for targeted policing are those that were, by precedent estimates, already over-represented in the historical police data. The freshly examined illegal acts that police document as a result of these directed patrols then feed into the predictive policing algorithm on following days, creating progressively more biased predictions. This generates a feedback loop where the model turn out to be gradually more self-confident that the places most likely to be subjected to further criminal activity are precisely the sites they had previously believed to be high in crime: "selection bias meets confirmation bias".

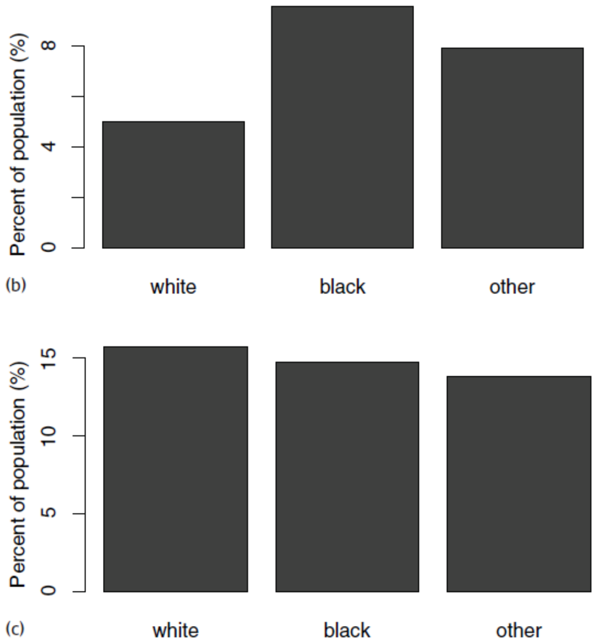

Figure 3.8: (b) Targeted policing for drug crimes, by race. (c) Estimated drug use by race (extracted from the original paper)

3.12 Amazon Doesn't Consider the Race of Its Customers. Should It? [25]

Year of publication: 2016
Domain: Services Providing
Discrimination problem: Many areas typically inhabited mostly by black people are not eligible for the same service offered in the surrounding white-populated areas

This article highlights the race discrimination caused, implicitly or explicitly, by Amazon when providing its same-day delivery program. Not every city and not every city area (identified by ZIP code) is eligible for that service, that is because the company made some business decisions in order to minimize costs associated with delivery. What emerged, as displayed in Figure 11, is that from many cities eligible for the same-day delivery were excluded from the service some areas predominantly populated by black people, agreeing to an assessment conducted by Bloomberg that compared Amazon same-day delivery areas with U.S. Census Bureau data. Figure 11. Percentage of Residents Eligible for Same-Day Delivery (extracted from the original paper). In Atlanta, Chicago, Dallas, and Washington, black citizens with access to Amazon same-day delivery are about half of the white ones which benefit of the service, even if both groups are living in neighbourhoods. “In New York City, same-day delivery is available throughout Manhattan, Staten Island, and Brooklyn, but not in the Bronx and some majority-black neighbourhoods in Queens” – we read in the paper. In some cities, Amazon same-day delivery extends many miles into the surrounding suburbs but is not available in some ZIP codes within the city limits. The most notable hole in Amazon’s same-day service is found in the city of Boston, where 3 ZIP codes covering the primarily black neighbourhood of Roxbury are excluded from the service, while it is available for the neighbourhoods that encircle it on all sides. The analysis showed that some excluded ZIP codes correspond with higher crime rates and any excluded areas have average household incomes below the national average. In those cities where the service was not ensuring to all the residents, those left out are disproportionately black. Amazon says the ethnic composition of neighborhoods is not part of the data examined when drawing up its maps, and its plan is to concentrate its same-day service on areas where there’s a soaring density of Prime members, that is a logical approach from a cost and efficiency perspective.

3.13 Machine Bias [6]

State: USA
Year of publication: 2016
Domain: Justice
Discrimination problem: Black people are more likely labelled as high-risk than white ones. Also they have more probability to be punished unjustly, and less probability to get away with it

This study conducted by ProPublica is about risk assessment algorithms use in the American justice system and it constitutes, as the authors say, “a part of a larger examination of the powerful, largely hidden effect of algorithms in American life”. It was investigated a commercial tool called COMPAS (which stands for Correctional Offender Management Profiling for Alternative Sanctions) developed by a for-profit company, Northpointe, that is one of the most popular scores used nationwide. COMPAS provides a score for each defendant ranged from 1 to 10, with ten being the highest risk. Scores 1 to 4 were labelled by COMPAS as “Low”; 5 to 7 were labelled “Medium”; and 8 to 10 were labelled “High.” What they found is that, not only the assessed scores were remarkably unreliable in forecasting violent crime, but also, they reflect significant racial disparities in forecasting the likelihood of an offender to be recidivist. Table 1 shows as black defendants were far more likely than white defendants to be wrongly judged at higher risk of recidivism, while white defendants were more likely than black defendants to be wrongly flagged as

	WHITE	AFRICAN AMERICAN
Labelled Higher Risk, But Didn't Re-Offend	23.5%	44.9%
Labelled Lower Risk, Yet Did Re-Offend	47.7%	28.0%

Table 3.1: False positives and false negatives rates for White and African American

low risk. In particular:

- Black defendants were often predicted to be at a higher risk of recidivism than they actually were. Black defendants who did not recidivate over a two-year period were nearly twice as likely to be misclassified as higher risk compared to their white counterparts (45% vs. 23%).

- White defendants were often predicted to be less risky than they were. White re-offenders within the next two years were mistakenly labelled low risk almost twice as often as black ones (48% vs. 28%).

- Even when controlling for prior crimes, future recidivism, age, and gender, black defendants were 45% more likely to be assigned higher risk scores than white defendants.

There is the same trend also with violent recidivism score:

- Black defendants were also twice as likely as white defendants to be misclassified as being a higher risk of violent recidivism. And white violent recidivists were 63% more likely to have been misclassified as a low risk of violent recidivism, compared with black violent recidivists.

- The violent recidivism analysis also showed that even when controlling for prior crimes, future recidivism, age, and gender, black defendants were 77% more likely to be assigned higher risk scores than white defendants.

Through the development of another statistical model, ProPublica was capable of estimate which are the most predictive factor of a higher risk score: defendants younger than 25 years old were 2.5 times as likely to get a higher score than middle aged offenders, even when controlling for prior crimes, future criminality, race and gender. As shown in Figure 3.9 race was also quite predictive of a higher score. While Black defendants had higher recidivism rates overall, when adjusted for this difference and other factors, they were 45% more likely to get a higher score than whites. Surprisingly, given their lower levels of criminality overall, female defendants were 19.4% more likely to get a higher score than men, controlling for the same factors.

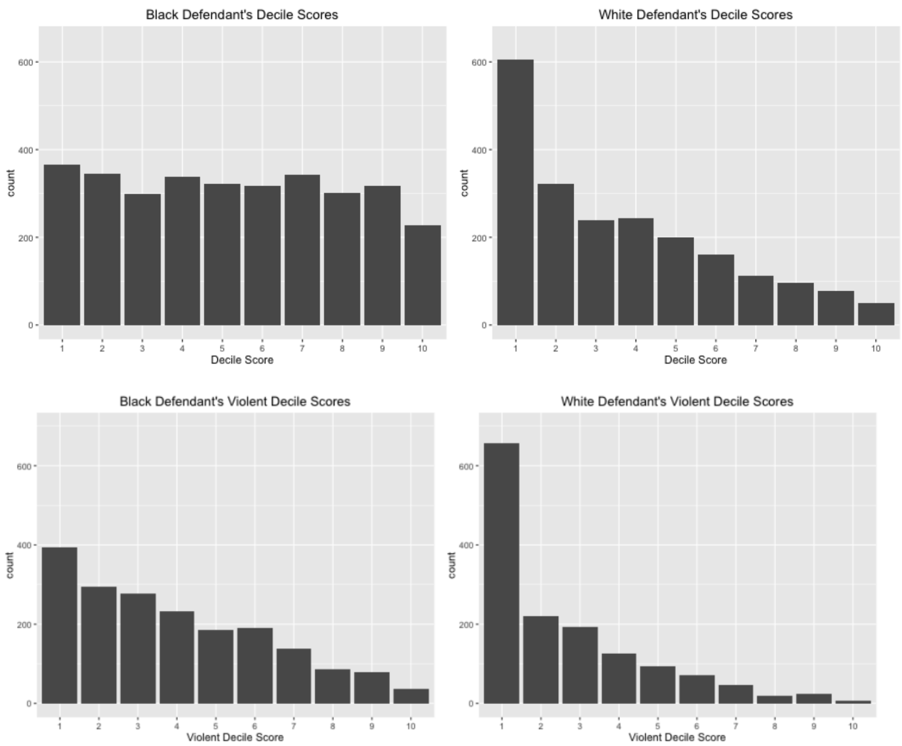

Figure 3.9: Scores Histograms (from original paper)

3.14 Discrimination through optimization: How Facebook's ad delivery can lead to skewed outcomes [26]

Year of publication: 2019
Domain: Social advertising
Discrimination problem: The advertising optimization algorithm exploits automatic classification to find the most suitable target for the ads, consequently excluding some people considered not interested

Due to the large variety of targeting features in online platforms, which may also be sensitive features such as user demographics and interests, researchers have raised concerns about discrimination in online advertising. Despite there are legal protections in the U.S. that prohibit discrimination against certain protected classes in advertising in the areas of credit, housing, and employment, researchers focused on Facebook and demonstrated that groups of users may be excluded from receiving certain ads because of the ads delivery system optimization, which is transparent to the advertiser. In practice, the Facebook Advertising platform, attempting to target the most receptive users for a given ad, may inadvertently cause ads to deliver primarily to a skewed subgroup of the advertiser's selected audience, producing an outcome that the advertiser may not have intended or be aware of. Ad delivery is affected as well by market effects and financial optimization that, because different desirability of user populations and unequal availability of users, may lead to skewed ad delivery. For example, the platform considers some user more "valuable" than other but may happen that this "valuable" user demographics are strongly correlated with protected classes. An advertiser who choose a low budget campaign is likely to never reach those users, ending up in fact with a discriminatory ad delivery. Even if there are no targeting features enabled, the ad delivery is skewed due to the content of the ad itself. For instances, ads that include any items that, according to stereotypes, would considered most interesting for men (e.g., bodybuilding) can deliver to over 80% of men, and those that include other items that would stereotypically be perceived as more interesting by women (e.g., clothes) can deliver to over 90% of women. Other differences in ad delivery can be significantly affected simply by the image. With the same stereotyped mechanism above, even if the ad's text and headline are misleading, the audience is selected considering as main factor the image's content alone. Furthermore, some experiments showed that every image is likely to be automatically classified, and for this reason the skew in ad delivery can be due in large part to skew in Facebook's automated estimate of relevance – image based, rather than ad viewers' interactions with the ad. Skewed delivery is observable also in employment and housing ads. Some ads for jobs in the lumber industry reach an audience that is 72% white and 90% male, some ads for cashier positions in supermarkets reach an 85% female audience, and ads for positions in taxi companies reach a 75% Black audience, even though the targeted audience specified is identical for all three. Despite the same targeting and budget, some of housing ads delivered to an audience of over 72% Black users, while others delivered to over 51% Black users.

3.15 Technical Reasons

3.15.1 Unbalanced data

It has been proven that problems of fairness and discrimination inevitably arise, mainly due to disproportionate datasets [3].The cause is probably given by how ML algorithms works. They analyse input data looking for recurrent patterns, and then they try to generalize results they found, to come out with codified knowledge, which is exploited to resolve future problems with new, previously unseen data. Having input data which are unbalanced, means causing representativity issues for minorities in the algorithm's outcome. If the initial data distribution is not obtained with the classical sampling methods, this problem causes underestimation or an overestimation of the groups. Many sampling techniques assume different constraints and requirements for how the samples are extracted. Their probability in fact must be known (and not null), and also the probability of extracting each same-length combination of observations must be equal. If the sampling process introduce any bias, it will propagate to estimates performed with that sample. It is clear why statistical sampling is a delicate and crucial step. As said before, many of the datasets used today for ADS have not been generated using probabilistic sampling, but are rather selected through non probabilistic methods. Often they come from some extraction process and/or manipulation of already existing databases used for users activity logging, historicization, etc. obviously the nature of the data stored is very depending on the nature of the activity itself, and may happen that - for certain activities in a certain context - some groups or individuals are more likely to be represented, others less. Furthermore, data directly recorded from every-day tasks and events that commonly happen in our society - bills payment, new hiring, online purchases, etc. - are likely soaked with the same stereotypes and inconsistencies our society is already affected by. Representativity is a property of the outcome of the extraction process, which itself has randomness as its property. Thus, samples which are non-probabilistic necessarily deserve particular attention and must be analysed in depth. Results which take into account demographic or statistical parity may be valid as well if the context does not require any special treatment for groups that are considered protected [27]. Finally, it is important to notice there is no any universally solution, but they vary according to both the nature and use of the data.

3.15.2 Bad quality

In computer science, "garbage in, garbage out" (GIGO) is a popular sentence to identify where "flawed, or nonsense input data produces nonsense output" [28] . The GIGO principle implies that the quality of the software is affected by the quality of the underlying data. As a consequence, computer generated recommendations or decisions are affected by poor input data quality. As a consequence, poor input data quality affects the decisions or predictions made by the software using that data, and implies ethical considerations on the confidence level of results, on the impact (in terms of relevance and scale) on people affected by the software decisions, and eventually even on the appropriateness of using that data at all. In the software engineering context, Data Quality is formally defined in the ISO/IEC 25012 Standard [29] as "the capability of data to satisfy stated and implied needs when used under specified conditions". The ISO/IEC standard defines 15 data quality characteristics, five of them inherent (quality depends only on the data per se: accuracy, completeness, consistency, credibility, currentness), three being dependent on the system in which data are used (portability, availability, recoverability), and the remaining are in the intersection of the two categories (accessibility, compliance, confidentiality, efficiency, precision, traceability, understandability). The 15 dimensions of data quality are operationalised by 63 metrics defined in the ISO/IEC 25024 Standard [30]. More specifically, we refer to inherent quality dimensions: accuracy, completeness, consistency, credibility, currentness. Recent research efforts [31, 32] showed that a measurement approach is effective in revealing data quality problems, especially for the inherent quality dimensions. Inherent quality measures are also more effective for purposes,because they are not affected by the context of use (e.g., hardware and software environment, computer-human interface). On these bases, it is reasonable to propose the ISO/IEC 25012 and 25024 standards models as a reference for quantitatively assessing the quality of data input and the consequential confidence and fairness of the software automated decisions made out of that data. According to the ISO/IEC 25024 standard the definitions of inherent quality dimensions are:

- **Accuracy:** Accuracy measures provide the degree to which data has attributes that correctly represent the true value of the intended attribute of a concept or event in a specific context of use.
- **Completeness:** Completeness measures provide the degree to which data associated with a target entity has expected values for all related properties of target entity in a specific context of use.
- **Consistency:** Consistency measures provide the degree to which data has attributes that are free from contradiction and are coherent with other data in a specific context of use. They can be either or both among data regarding one target entity and across similar data for comparable target entities.
- **Credibility:** Credibility measures provide the degree to which data has attributes that are regarded as true and believable by users in a specific context of use.
- **Currentness:** Currentness measures provide the degree to which data has attributes that are of the right age in a specific context of use

3.15.3 Bad use

In many classification tasks the ML algorithm is trained with data containing variables which are sensitive for the individual. These variables represent characteristics of the individuals which may identify them in certain *protected categories* dependently on the context in which the ML algorithm is used. To allow the statistical model know about these characteristic and letting it performing predictions and classifications based on them implies deep consequences on our analysis. It may happen our model learn to discriminate against protected categories of individuals, which is exactly the situation we want to avoid. One may think that removing or ignoring sensitive variables could be helpful to obtain impartial results, but unfortunately this operation not only end up to be useless also it may causes additional issues. On a typical dataset there are usually many features which are correlated with the sensitive attribute. Even if the level of correlation may be barely evident, when codified all together a large number of correlated features may be sufficient to successfully identify sensitives attributes. For this reason a classifier trained without sensitive features will have the same performance of the one trained using them.

Chapter 4

Algorithmic Fairness

When we leave to software and algorithms the capability of deciding for us, or taking decisions that deeply impact our lives, maybe in a way that is completely-transparent, we would like those decisions were rightful or at least *fair*. Machine learning algorithms and statistical models in this sense promise to satisfy our need of reliability by introducing data-driven approach. From AI algorithms we expect they are able to learn, to generalize from the specific examples we provide them, and solve future unseen problems. That is the key concept of machine learning. Many experiments proved that results obtained using machine learning methods are often more accurate of the ones gathered from years-experienced professional [33]. Furthermore ML is suppose to be impartial, faster, and capable of uncovering factors which may be relevant but as complex as humans usually overlook them. As we have seen in the previous chapter many complications arise when examples we provide reflect historical prejudices against certain social groups, prevailing cultural stereotypes, and existing demographic inequalities. Almost surely finding patterns in these data will mean replicating these very same dynamics. Without specific intervention, machine learning will extract stereotypes, including incorrect and harmful ones, in the same way that it extracts knowledge [5]. Once we observe such inequalities and disparities we can not say with certainty that designers and developers of the algorithm intended to make them arise. Furthermore it is not immediate to say when the observed inequalities can be considered act of discrimination without having sociological and philosophical tools, and a technical mathematical formalization of the problem as well. Concepts such as discrimination and fairness have long been studied and debated from moral and political philosophers thus it should not be surprising that attempts to formalize fairness in ML contain echoes of these old philosophical debates [34]. However it is clear that *algorithmic discrimination* is something very different from the classical form of discrimination, thus it needs a different approach to be investigated and confronted. In fact it is relative simple to trace back reasoning and characteristics on which is based the classical form of discrimination, such as bad intentions, animosity, humiliation and lack of respect against individual of certain group, gender, or ethnicity. The decision-maker's intent seems to be the key to discrimination. At the contrary when speaking about algorithms we can not talk about thinking or intentions, so identifying the hallmarks of discrimination become quite challenging.

4.1 Fairness criteria in supervised learning

Despite the lack of generally valid assumptions on which we should consider fair when talking about algorithmic decisions outcome, the community moved so far as defining different fairness criteria. These criteria identify and formalize - especially in the field of classification done by ML algorithms - non-discriminating behaviors. Mostly of them are well summarized by Barocas et al. (2018), who categorized many definitions of fairness appeared in the past - under different shades - into three macro areas. The proposed criteria are expressed in function of the joint distribution of the sensitive attribute A, the target variable Y, and the classifier R.

Independence: simply requires the sensitive characteristic to be statistically independent of the score.

Definition: *The random variables (A, R) satisfy independence if $A \perp R$.*
The definition above simplifies in the case of binary classification:

$$\Pr\{R = 1|A = a\} = \Pr\{R = 1|A = b\}, \forall\, groups\; a, b$$

Separation: Correlation between the score and the sensitive attribute is allowed to the extent that is *justified by the target variable.*.

Definition: *The random variables (R, A, Y) satisfy separation if $R \perp A|Y$.*
The definition above in the case of binary classification is equivalent to:

$$\Pr\{R = 1|Y = 1, A = a\} = \Pr\{R = 1|Y = 1, A = b\}, \forall\, groups\; a, b$$

$$\Pr\{R = 1|Y = 0, A = a\} = \Pr\{R = 1|Y = 0, A = b\}, \forall\, groups\; a, b$$

Sufficiency: requires that the score already subsumes the sensitive characteristic for the purpose of predicting the target.

Definition: *The random variables* (R, A, Y) *satisfy sufficiency if* $Y \perp A|R$.
The definition above simplifies in the case of binary classification:

$$\Pr\{Y = 1|R = r, A = a\} = \Pr\{Y = 1|R = r, A = b\}, \forall \, groups \; a, b, \forall \, r \in R$$

Application of the criteria: there are principally three ways of applying the criteria above, each one focus on a specific time of the algorithm life cycle:

- Pre-processing: manipulate data and adjust the feature space before feeding the algorithm with the data
- Training time: introduce constraints into the optimization process of the statistical model
- Post-processing: after the model is build, which means the training phase was completed, adjust the outcome appropriately

Each method presents different pros and cons. The main advantage of pre-processing is that it is generally agnostic to what will happen in the new feature space in the following phases of the algorithm. It means we do not need to know which model will be used, in any case the transformation performed in this step will be propagated to the other ones. Applying a criterion during the training time often means reaching the best performance in fact we perform directly on the model optimization process. This method assumes we have access to the raw data and training pipeline and causes some loss of generality, since we adjust the algorithm for a specific model. In the case of post-processing we get a derived model adjusting the original one (i.e. adding random noise, modifying weights of the sensitive attributes). We get rid of the pipeline's complexity, since we do not need re-training and we will work regardless of the model detail. However this approach may be less effective than others.

4.2 Fairness in Ranking systems

If the fairness criteria indicated above are specifically thought for classification task in supervised learning, the field of fairness for rankings has been a relatively under-explored domain despite the growing influence of online information systems on our society and economy [12]. Existing works on fairness in ranking mainly focus on a sufficient presence, a consistent treatment and a coherent representation of different groups across each ranking positions [35]. Many of those works are focused on development of a fairness-aware ranking given a set of scores, and can be considered methods for post-processing results, where they are given a ranking and re-sort elements to reached a desired result. Yang and Stoyanovich [36] proposing definitions and methods that minimize the difference in the representation between protected and non-protected groups introducing a generative model for fair rankings. Zehlike et al. [37] design a statistical test for the generative model of Yang and Stoyanovich [36]. Celis et al. [38] examine a scenario in which many protected groups are present and hence several vectors containing one protected elements (one per group) at each position are given as input. Joachims and Singh [39] first introduced the definition of *exposure* of a group, which explains how the probability of a user sees an item ranked at a certain position, decreases rapidly with the position. Our work, described in the following chapter, is based on distributive justice theory of Roemer [40] and a methodology proposed by Brunori et al. [41]

Chapter 5

Methodology

In the head of our aims there is the willing of experiment with interdisciplinary concepts and see if they could be useful in order to build models that are fair and human-centered. In particular we investigate how to combine methods from the philosophical, legal and economic sciences into algorithmic systems, not only to avoid adverse outcomes such as discriminatory behaviors, but also to foster positive effects on society and reduce social inequalities. For this purpose, we assume as fundamental references in the philosophical-legal and economic fields the distributive justice and Equality of Opportunity (EOP) studies, which aim to establish a theory of social justice based on the reallocation of resources. In this thesis we propose a hypothetical scenario of a selection process in which a finite number of students have to be chosen on the basis of their personal performance at school, so as to reward the most deserving. We imagine the solution of this task may looks trivial and many traditional ranking systems may perform their evaluation simply sorting all the candidates on their final average score. But from a really meritocratic point of view the selection process is far more complicated than this. For example is not easy to take the student who worked hardest, or the one who learned most, simply because the final score does not reflect the starting conditions of the students neither their educational background. Someone may have been helped with his homework by very careful parents, others instead may have been distracted by his numerous young brothers. These ones are reasons independent from the control of the student, from the effort he put on the study, and from his capacity of learning and being a valid student. The methodology we follow aim to overcome this and other problems. We analyze the students' performance and build our ranking trying to bring justice to most penalized students. We try to take our selection by putting every student in the same starting conditions, in this way we want to guarantee to everyone an opportunity of being elected which is based on their real individual capacity and nothing else. We examine the trade-off of the expected outcome for groups of individuals in the ranking system before and after the application of our distributive fairness approach; finally we explore the trade-off of Equality of Opportunity in the different rankings performed.

5.1 Assessing Distributive Fairness

5.1.1 Equality of Opportunity: a machine learning approach

The idea of equality of opportunity formalized by Roemer [42] in obtaining well-being is based on the basic principle that the individual's achievement should depend on choice, effort, and ability, not on the circumstances of birth. The theory is based on four key principles: circumstances, effort, responsibility and reward. The first assumption that Roemer formulates on the idea of equality is referred to the so-called *principle of compensation*. He claims that if inequalities in a set of individuals are caused by birth circumstances, which include variables such as gender, race, or family socio-economic status and so forth, then these are morally unacceptable and must be compensated by society. The second assumption is based instead on individual utility, or well-being, in relation to individual responsibility, also called the *principle of responsibility*. In fact, he argues that in determining results, in addition to the circumstances of birth, the effort that individuals invest in achieving the acts they perform and for which they are fully responsible also holds a key role. Therefore, a society that guarantees equal opportunities is a society in which results, well-being, or utility, are distributed independently to circumstances, and in which individual responsibility and effort are fully recognized. According to Roemer's general theory of EOp, policies should be oriented to equalize the opportunities that different *types*, or groups of individuals, categorized in accordance with diverse circumstances, must be able to have in order to achieve a given goal. A *type* is a set of individuals sharing the same circumstances, while the set of individuals characterised by the same degree of effort is called a *tranche*. The reason why equality of opportunity is mainly associated with the name of Roemer is due to the fact that he did not only embrace and clarify its theoretical and conceptual framework, but he was the first to propose an operational algorithm that gave rise to an interesting empirical literature to which he contributed significantly. A first distinction between the various nuances deriving from the literature concerns the partitioning of individual characteristics into two categories, effort and circumstances. Explaining the differences in the various theories is beyond the scope of this work; for our purpose it is sufficient to point out that different partitions correspond to different notions of EOp.

More generally, the statistical approach suggested by Roemer to measure equality of opportunity is valid for any nuance of the theory. He assumes that each individual outcome y can be expressed as the result of a combination of effort e ($e_i \in \Phi$, where

Φ is the set of all possible level of effort) and circumstances c ($c_i \in \Omega$, where Ω is the set of all possible circumstances); the individual outcome is therefore produced by the function $g : \Omega \times \Phi \Rightarrow \mathbb{R}$ such that:

$$y_i = g(c_i, e_i) \tag{5.1}$$

The model presented is a purely deterministic model in which measurement errors or random components are neglected, as suggested by several authors [43], [44], [45], [46]; this problem is due to the fact that effort (e) is not a directly observable datum, as well as the g function. To overcome some problems Roemer supposes that the g function is fixed and identical for each individual and introduces two basic hypotheses:

Hypothesis 1 (H1)**.** *The g function is monotonically increasing in effort (while subjective utility is commonly considered decreasing in standard notions of effort).*

Hypothesis 2 (H2)**.** *The distribution of effort is independent of circumstances.*

We will resume the treatment of the hypotheses thus formulated in the following Sections (5.1.1, 5.1.1).
A second differentiation in the different approaches for the estimation of EOp is related to the partitioning of individuals into *types* and *tranches*.

$$M_{type,effort} = M_{i,j} = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,j} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,j} \\ \vdots & \vdots & \ddots & \vdots \\ m_{i,1} & m_{i,2} & \cdots & m_{i,j} \end{pmatrix}$$

For the *ex ante approach*, or *type-compensation principle*, EOp occurs if the set of opportunities of different individuals is identical, independently from circumstances. Roemer states that *"it is good to transfer from an advantaged type to a disadvantaged type, provided that the ranking of types is respected. Suppose that between two types, one is unambiguously better off than the other, that is, the outcomes can be ranked unambiguously according to first-order stochastic dominance. Then a transfer from the dominant type to the dominated type for some effort level, ceteris paribus, is EOp enhancing"*[47]. The type approach focuses on differences in the perspectives of ex ante outcomes for classes of individuals with identical circumstances, thus focusing on inequalities between types and being neutral towards inequalities within types.
For the *ex post approach*, or *tranche-compensation principle*, EOp occurs if all those who spend the same level of effort achieve the same result. Roemer states that *"the closer each column is to a constant vector, the better. If for some effort (column), the inequality of outcome across types is reduced, and everything else*

remains unchanged, EOp has been improved"[47]. In contrast to the type approach, the tranches approach focuses on ex post inequalities in classes of individuals with the same degree of effort. Consequently, the approach focuses on the distribution of inequality in outcomes within tranches.
Roemer's definition of EOp can therefore be summarized in the following model: let a population of $1...N$, individuals i, with an outcome y_i, assigned to a finite set of types $t = 1...T$. Let f^t be the fraction of the population of type t. Let an objective be given, i.e. a threshold set by the decision maker to reach EOp. The value of the degree to which an individual achieves an objective is a function of circumstances, effort and social policy θ ($\theta \in \Theta$, where Θ is the set of social policies):

$$u^t(e_i, \theta), \tag{5.2}$$

where u^t is the average achievement of the objective in type t that spend effort e when the policy is θ. Let $G^t_\theta(e_i)$ be the function of effort distribution in type t when the policy is θ. Therefore, with the available set of data $T, G^t_\theta(e), f^t, u, \theta$ we can then rewrite the equation (5.1) in this way:

$$y_i = G^t_\theta(e_i) \tag{5.3}$$

Circumstances and Types

The identification of types and effort design requires society to have at least a similar, if not unified, view of how to distinguish actions and variables that belong to the sphere of individual responsibility or circumstances; a unique approach to this diversification ensures a unique understanding of the results arising from the measurement of equal opportunities. Roemer's approach to measuring inequality of opportunity involves considering a situation as unequal if two individuals who have both made the same choices and had different birth circumstances, have obtained a different outcome. The first step to make Roemer's method effective is to identify types, i.e. to identify the combinations of the realization of the circumstances that partition the population into N subsets, in which each individual is included once and only once. The simplest empirical methodologies identify types on the basis of socio-economic uniform features, such as gender, ethnicity, income, and compute the value of opportunities according to the outcomes obtained by the individuals belonging to each type. Many AI/ML systems actually adopt this methodology to achieve a fairness result; the definition of discriminating circumstances is made on the basis of a historical discrimination that has led individuals belonging to these minority categories to be in a disadvantaged position [48]. Minority categories are therefore defined by identifying variables or proxy variables of real discrimination, and these variables, such as gender, ethnicity, place of birth, are called protected or sensitive attributes [49]. This kind of approach actually hides considerable methodological problems in the correct identification of types. Although straightforward and simple, the method described above does not allow to take into consideration all those variables that contribute to shaping both the responsibility of the individual and the circumstances of birth. In general, Roemer does not address the problem of identification of types and circumstances, but over the years several important empirical contributions have been provided to trace the structure of the method. Some of the most relevant are the inferential conditional trees proposed by Hothorn et al. [50], the non-parametric method by Checchi and Peragine [51] and latent class models by Li Donni et al. [52]. It is beyond the scope of our work to analyse and discuss the trade-offs between the various methodologies proposed, therefore we focus only on the Hothorn methodology that we found most effective in determining types. (To the best of our knowledge) To the best of our knowledge, the sole work involving the algorithm proposed by Hothorn was applied by Brunori and Neidhöfer [53] to study socio-economic differences on panel data. In its general meaning, the algorithm for the determination of types exploits the permutation test theory developed by Strasser [54] to generate recurring binary partitions overcoming the problem of overfitting and variable selection. In fact, recursion takes advantages of the conditional distribution of statistics that measure the correlation or association between the response variable and its covariates and performs multiple hypothesis tests to determine the significance of the correlation or association; if it is not possible to identify a statistically significant correlation or association between the response variable and any of the covariates, recursion stops. In the algorithm we have implemented we use conditional inference trees to recursively partition the Euclidean space of the variables of the individuals in convex sets of hyperplanes. The convexity of sets is a fundamental property of this methodology because it allows us to affirm that individuals belong to one and only one subset, and therefore to one and only one type. We briefly describe below the steps of Hothorn's algorithm for conditional recursive inference trees to perform the identification of Roemer types.

Given a response variable Y and a set of covariates $X(x_1, ..., x_m)$ we assume that the conditional distribution of the response variable $P(Y|X)$ given the covariates is a function f of the covariates such that $P(Y|f(X))$. At each step the algorithm tests the partial null hypothesis of independence $H^0_{partial} : P(Y|X) = P(Y)$ between the response variable and any of the covariates, and stops if the hypothesis cannot be rejected at a certain level of α[1] previously selected; otherwise, it selects the covariate

[1]The value of α controls the probability of falsely rejecting H_0 at each node, and its use is the same to conventionally control Type I and Type II errors in hypothesis tests [50].

x_M with the highest correlation or association to Y through the *Simple Bonferroni-adjusted P-values*[2] that indicates the deviation from the partial hypothesis $H^0_{partial}$. The test is performed on each covariate to test the global null hypothesis. At the end of the procedure a set of N types is obtained as in figure 5.1 shaped after the execution of the multiple independence tests on each circumstance of individuals.

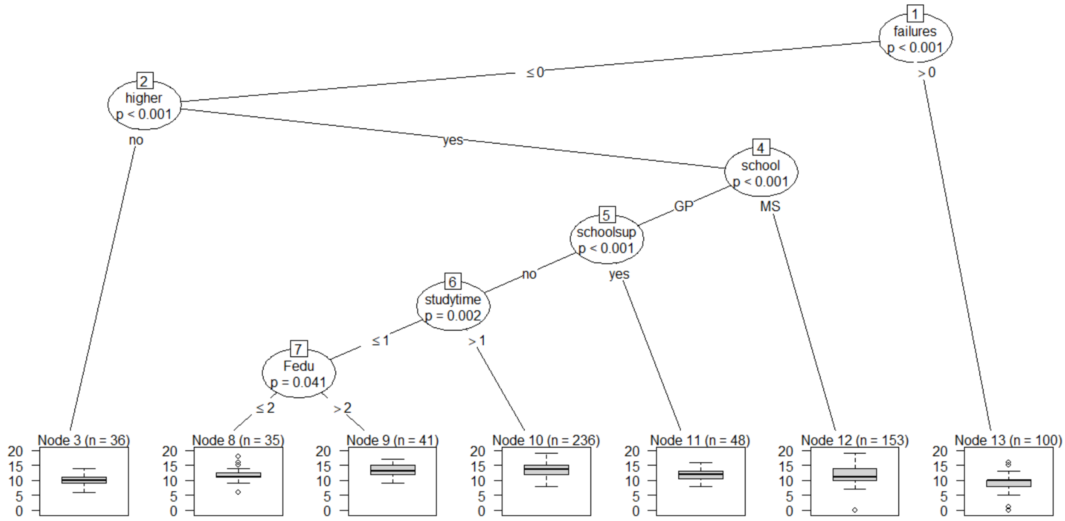

Figure 5.1: Conditional inference tree for types estimation

Circumstances and Effort

To discuss the effort estimate, we resume the assumptions HP1 and HP2 expressed in Section 5.1.1. Although the first hypothesis does not present particular problems, the second one poses more issues. Individuals with more advantageous circumstances may consequently be more inclined to exert a greater degree of effort. In any case, it would not be quite possible to assign to an individual the accountability of his or her level of outcome if the degree of effort depended on exogenous circumstances. Hence, from a computational point of view, estimating effort is one of the most complex aspects, as its difficulty in being observed is the result of a process of maximizing individual preferences. Since we assume that the effort is not directly observable, it is necessary to deduce its value from observable behaviours, i.e. a

[2]Use *t tests* to make pair comparisons between group means, but check the overall error rate by setting the error rate of each test to the experimental error rate divided by the total number of tests. In this way, the level of significance observed is adjusted considering multiple comparisons are being performed (For further details see Bonferroni [55])

proxy measure is needed to measure and compare the effort of different individuals. The definition and measurement of effort by Roemer has changed over time; the definition to which we refer considers the relative individual effort determined not only by the variable of preference (the degree of effort); on the contrary is determined by all the elements that establish the location of each individual in the distribution of the advantages that characterizes the given type. Reomer argues that it exists an effort distribution function that characterizes the entire subgroup within which the location of the individual is set and what is needed is a measure of effort that is comparable between different types. The assumption at the basis of this assumption is that two individuals belonging to a different type t who occupy the same position in their respective distribution functions have exerted the same level of effort - and therefore of responsibility -. Since, under the same circumstances, individuals who make different choices exercise different degrees of effort - and thus achieve a different outcome -, the differences in outcome within the same type are by definition determined by different degrees of effort, and therefore are not considered in the computation of the EOp. In general, Roemer states that to estimate effort it is necessary to aggregate individuals according to their circumstances (see **type estimation** in Section 5.1.1), compare outcome distributions and measure the degree of effort an individual has exerted using the quantile he or she occupies in his or her type distribution. Since for HP1 the outcome function is monotonous and for HP2 the effort is orthogonal to the circumstances, it is possible to measure the effort of an individual belonging to a generic type by the rank or quantile of the effort distribution in which that individual is positioned. Therefore, all the individuals positioned at the same quantile in the distribution of the respective type are by assumption characterized by the same level of effort. As we have highlighted in Section 5.1.1, the *ex-ante* and *ex-post* approaches express two different methods of achieving EOp. Hereafter we will refer to the *ex-post* approach, or *tranche-compensation principle*, which is the methodology we adopted.
Let the tranche vector $Y_{t,\lambda}$ be the set of outcomes enclosed in a given quantile λ of a type t; it expresses the different outcome values of individuals who exercised the same degree of effort. Since the inequality in outcome within $Y_{t,\lambda}$ is not explained by this methodology, several papers propose to apply a smoothing function to eliminate this unexplained inequality (Checchi and Peragine, Brunori and Neidhöfer). The standardized distribution of the outcome of individual i belonging to type t and located at quantile λ, is obtained by scaling each tranche until all have the same mean of the total distribution, and is expressed by the following equations:

$$y^t(G^t_\theta(e)) = y^t(\lambda) \quad \Rightarrow \quad F^t(y) \vdash y^t(\lambda), \tag{5.4}$$

where $F^t(y)$ is the cumulative distribution of outcomes in type t,

$$\tilde{y}_i^t(\lambda) = y_i^t(\lambda)\frac{\mu}{\mu^{\lambda}} \;\; \Rightarrow \;\; \tilde{F}^t(y) \vdash \tilde{y}_i^t(\lambda), \tag{5.5}$$

where $y^t(\lambda)$ is the outcome of individual i in type t at given quantile λ, derived from the cumulative distribution of the type-specific cumulative distribution in equation 5.4, μ is the mean of population's outcome, μ^{λ} is the mean of individual's outcome located at quantile λ over all types t.
In this way, observed inequalities are exclusively due to circumstances or degrees of effort; therefore, only inequalities resulting from exogenous circumstances are observed and not those arising from the responsibility of individuals. As Brunori and Neidhöfer [53] suggests, for the smoothing process we adopt one of the proposed Bernstein's polynomial approximation application (Leblanc, Zhong)to obtain the standardized distribution of tranche vectors $Y_{t,\lambda}$. The methodology is described below.
The outcome of individuals y can be considered as a sequence of random variables having a density function f supported by a closed interval $[a, b]$ and a cumulative distribution function F, where $y \in [a, b]$ and y is a positive continuous variable. The continuous density function f defined on $[a, b]$ can be approximated by a linear combination of Bernstein's polynomial bases of degree m, defined by the formula:

$$\tilde{f}_m(y) = \mathbb{B}_m(y, a, b) = \sum_{i=0}^{m} f(\frac{i}{m})b_{i,m}(y, a, b), \quad a \leq y \leq b \tag{5.6}$$

where $b_{i,m}(y, a, b)$ are binomial probabilities defining the Bernstein basis polynomials in generalized polynomial space:

$$b_{i,m}(y, a, b) = \frac{1}{(b-a)^m}\binom{m}{i}(y-a)^i(b-t)^m - i, \quad \forall i = 1, ..., m \tag{5.7}$$

The cumulative smoothed distribution of the outcome for type t $F^t(y)$ [Equation 5.4] with Bernstein's approximation is simply derived by estimating the density function for each type t, by approximating each function with Bernstein polynomials, and than by computing the integral function of $\tilde{f}_m(y)$:

$$F^t(y) = \int_a^b \tilde{f}_m^t(y)\, dy \tag{5.8}$$

To determine the degree of the polynomial that best approximates the function $\tilde{f}_m(y)$, we use the degree of the polynomial that maximizes the out-of-sample LogLikelihood by ten-fold cross-validation, as suggested by Brunori and Neidhöfer [53].

5.2 Policy

We adopt three different policies borrowed from distributive justice area [58] in order to perform our ranking.

- **Equity:** *"Members' outcomes should be based upon their inputs. Therefore, an individual who has invested a large amount of input (e.g. time, money, energy) should receive more from the group than someone who has contributed very little."*[58]
- **Equality:** *"Regardless of their inputs, all group members should be given an equal share of the rewards/costs"*[58]
- **Need:** *"Those in greatest needs should be provided with resources needed to meet those needs. These individuals should be given more resources than those who already possess them, regardless of their input."*[58]

For each ranking performed with a given criterion we take the top 100, 250, and 500 individual realizing a total of 9 different ranking.

5.2.1 Equity

Ranking based on equity's policy is performed following the algorithm 1.

Algorithm 1 Equity rank
input: dataset D, size k
output: ordered list of k rows

$sorted_D \leftarrow$ D ordered descending on standard outcome
$sorted_{sets} \leftarrow$ split $sorted_D$ on 50 sequential subsets
for all $set \in sorted_{sets}$ **do**
    for all $types \in set$ **do**
        Compute *mean outcome* and assign it to each individual of that type
    end for
    Compute *mean outcome* and assign it to each individual of that *set*
end for
$merge_D \leftarrow$ Merge all *sets*
return k *rows* $\in merge_D$ sorted on last computed *mean outcome*

5.2.2 Equality

Ranking based on equality's policy is performed following the algorithm 2. We choose to ensure equality for men and women so we based on *Sex* attribute.

Algorithm 2 Equality rank
input: dataset D, size k
output: ordered list of k rows

$sorted_D \leftarrow$ D ordered descending on outcome
$sorted_{sets} \leftarrow$ group $sorted_D$ by Sex attribute
for all $set \in sorted_{sets}$ **do**
    Sort *set* descending on *outcome*
end for
return k *rows* from each group

5.2.3 Need

Ranking based on need's policy is performed following the algorithm 3. We choose to ensure equality for men and women so we based on *Sex* attribute. It combines approaches based on equity and equality.

Algorithm 3 Need rank
input: dataset D, size k
output: ordered list of k rows

$sorted_D \leftarrow$ D ordered descending on standard outcome
$sorted_{sets} \leftarrow$ split $sorted_D$ on 50 sequential subsets
for all $set \in sorted_{sets}$ **do**
    for all $types \in set$ **do**
        Compute *mean outcome* and assign it to each individual of that type
    end for
    Compute *mean outcome* and assign it to each individual of that *set*
end for
$merge_D \leftarrow$ Merge all *sets*

$sorted_{sets} \leftarrow$ group $merge_D$ by Sex attribute
for all $set \in sorted_{sets}$ **do**
    Sort *set* descending on *outcome*
end for
return k *rows* from each group

5.3 Metric

5.3.1 Inequality

In order to compute inequality of opportunity, an inequality index applied to the standardised distribution Y derived from the equation 5.5 must be employed. The measurement of inequality of opportunity can be treated as a two-stage process:

1. the actual distribution of Y is transformed into a counterfactual distribution $\tilde{Y}$ which expresses the unfair inequality in Y because it is due to exogenous circumstances, while all the fair inequality due to individual responsibilities is removed;

2. secondly, a measure of inequality is applied to $\tilde{Y}$.

However, computing the equation 5.5 means getting an outcome vector in which the only inequality expressed is that within the tranches: an inequality index applied to this distribution captures exclusively and completely the outcome inequalities resulting from the circumstances, i.e. inequality of opportunity.
For this purpose we use the Gini index, a statistical concentration index that measures the degree of inequality of a distribution, commonly used to measure the distribution of income. The index lies in a range between 0 and 1; a low or equal to zero Gini index indicates the tendency to the equidistribution and expresses perfect equality; on the contrary, a high or equal to 1 value indicates the highest concentration and expresses the condition of maximum inequality. The Gini index calculus is based on the Lorenz curve of the distribution[3] (Figure 5.2).

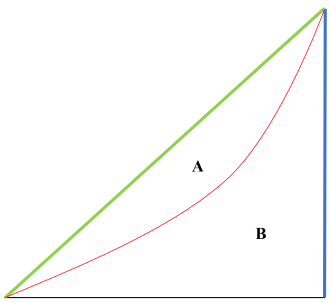

Figure 5.2: Graphical representation of the Gini index through Lorenz curve

The blue line represents the line of perfect inequality, the green line represents the line of perfect equality, or line of equidistribution, and the red line is the Lorenz

[3]For further details on Gini index and Lorenz curve calculus see Lorenz [59], Gini [60] and Gastwirth [61]

curve. The area A between the lines of perfect equality and the Lorenz curve is called the concentration area and represents the deviation from perfect equality; Gini's index is the ratio between the area A and the total area:

$$GiniIndex = \frac{A}{A+B} \tag{5.9}$$

The inequality of opportunity through the application of the Gini index is therefore expressed by the following equation:

$$InequalityofOpportunity = GiniIndex(\tilde{Y}) \tag{5.10}$$

5.3.2 Diversity

The measure of diversity indicates abundance or lack of species in a given population [62]. We use Shannon index which is one of the most popular in literature.

$$H = \sum_{i=1}^{R} p_i \ln p_i$$

R identifies how many different types the dataset contains, and p_i is the frequency of the type i^{th}

5.3.3 Entropy

The Theil index is mainly used to estimate economic inequalities [63]. The concept of entropy of a system (our dataset) can be summarized as following: *"In a system a certain amount of transformations is possible. The sum of transformations, which already have occurred, cannot be reversed without help from outside. Entropy is a measure for how many transformations already have occurred in that system. The redundancy serves as a measure for how many transformation opportunities still are available. If completely equal distribution (of whatsoever) in a system leads to maximum entropy of that system and if low entropy of that system is caused by high distributional inequality, then achieving equal distribution means that the distribution process is saturated."* [64]

5.3.4 Opportunity-Loss Profile

Opportunity-Loss Profile (OLP) indicates in a certain distribution which population types are disadvantaged or advantaged with respect of their outcome, before redistribution (BR) and standardized outcome, after redistribution (AR). Computation involves across different quantiles the estimation of mean outcome and mean standardized outcome inside each population type. The estimation is performed once for each quantile and the result is stored. The type more often appears as the one with the minimum outcome is classified as disadvantaged. The type more often appears with the maximum outcome is classified as advantaged.

Algorithm 4 Opportunity-Loss Profile
input: dataset D
output: list of 4 items

for all *quantile* $\in$ *quantiles* **do**
    for all *type* $\in$ *types* **do**
        Compute *mean outcome* between individuals of *type*
        Compute *mean standard outcome* between individuals of *type*
    end for
    extract disadvantaged type BR taking the *min(mean outcome)*
    extract disadvantaged type AR taking the *min(mean standard outcome)*
    extract advantaged type BR taking the *max(mean outcome)*
    extract advantaged type AR taking the *max(mean standard outcome)*
end for
return *types* extracted more frequently across quantiles

5.3.5 Opportunity-Loss Rate

Opportunity-Loss Rate (OLR) indicates for each type the extent of outcome variation after the redistribution process. Types with negative values have lost their outcome following redistribution. Types with positive values have increased their outcome following redistribution.

Algorithm 5 Opportunity-Loss Rate
input: dataset D
output: dataset D **with an additional column**

for all *type* $\in$ *types* **do**
    $om \leftarrow mean(type\$outcome_{before})$
    $oms \leftarrow mean(type\$outcome_{after})$
    $type\$opportunity_{loss} \leftarrow om - oms$
end for
$normalize(D\$opportunity_{loss}, -1, 1)$
return D

5.3.6 Distributive Rate

Distributive Rate indicates for each individual the extent of outcome variation after the redistribution process. Individual with negative values have lost their outcome following redistribution. Individual with positive values have increased their outcome following redistribution.

Algorithm 6 Distributive Rate
input: dataset D
output: dataset D **with an additional column**

for all $student \in D$ **do**
    $om \leftarrow mean(type\$outcome_{before})$
    $oms \leftarrow mean(type\$outcome_{after})$
    $student\$distributive_{rate} \leftarrow om - oms$
end for
$normalize(D\$distributive_{rate}, -1, 1)$
return D

5.3.7 Reward Profile

Reward Profile (RP) indicates in a certain distribution which population types are disadvantaged or advantaged with respect of their outcome evaluated before the policy application (BP) and after the policy application (AP).

Algorithm 7 Reward Profile
input: dataset D
output: list of 4 items

for all $quantile \in quantiles$ **do**
    for all $type \in types$ **do**
        Compute *mean outcome BP* between individuals of *type*
        Compute *mean outcome AP* between individuals of *type*
    end for
    extract disadvantaged type BP taking the *min(mean outcome BP)*
    extract disadvantaged type AP taking the *min(mean outcome AP)*
    extract advantaged type BP taking the *max(mean outcome BP)*
    extract advantaged type AP taking the *max(mean standard outcome AP)*
end for
return *types* extracted more frequently across quantiles

5.3.8 Reward Rate

Reward Rate (RR) indicates the extent of outcome variation after the application of the redistribution's policy for each type. Types with negative values have lost their outcome following redistribution. Types with positive values have increased their outcome following redistribution.

Algorithm 8 Reward Rate
input: dataset D
output: dataset D **with an additional column**

for all $type \in types$ **do**
  $om \leftarrow mean(type\$outcome_{beforepolicy})$
  $oms \leftarrow mean(type\$outcome_{afterpolicy})$
  $type\$reward_{rate} \leftarrow om - oms$
end for
$normalize(D\$reward_{rate}, -1, 1)$
return D

Chapter 6

Study Case

Our analysis with the methodology discussed in the previous chapter is based on the Student Performance Data Set [65]. We imagine an hypothetical scenario where high school students compete to gain access to "closed number" university. We assume university is equipped with a ranking system that fill a finite number of available positions based on some students' characteristic. The standard system must select the best candidates by evaluating only their performance and nothing more. Our ranking system aims to guarantee the Equality of Opportunity between candidates analyzing their circumstances and their effort in order to extract the most deserving of being admitted to university. We chose this scenario for two reasons: first because of its relevance in the modern universities, secondly because the *numerus clausus* method was historically cause of discrimination against some ethnic groups and religions [66].

6.1 Dataset

The dataset we use comes from data collected from two public schools of the Alentejo region of Portugal, during 2005-2006 school year. The information contained are based on paper sheets report and questionnaires containing lots of demographic, social, emotional, and school related questions. Finally we report the list field contained in the dataset (see A.1 for further details):

Attribute	Description (Domain)
sex	student's sex (binary: 'F' - female or 'M' - male)
school	student's school (binary: 'GP' or 'MS')
age	student's age (numeric: from 15 to 22)
address	student's home address type (binary: 'U' - urban or 'R' - rural)
famsize	family size (binary: 'LE3' - less or equal to 3 or 'GT3' - greater than 3)

Pstatus	parent's cohabitation status (binary: 'T' - living together or 'A' - apart)
Medu	mother's education (numeric: 0 - none, 1 - primary education, 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education
Fedu	father's education (numeric: 0 - none, 1 - primary education, 2 - 5th to 9th grade, 3 - secondary education or 4 - higher education
Mjob	mother's job (nominal: 'teacher', 'health' care related, civil 'services', 'athome' or 'other')
Fjob	father's job (nominal: 'teacher', 'health' care related, civil 'services', 'athome' or 'other')
reason	reason to choose this school (nominal: close to 'home', school 'reputation', 'course' preference or 'other')
guardian	student's guardian (nominal: 'mother', 'father' or 'other')
traveltime	home to school travel time (numeric: 1 - <15 min., 2 - 15 to 30 min., 3 - 30 min. to 1 hour, or 4 - >1 hour)
studytime	weekly study time (numeric: 1 - <2 hours, 2 - 2 to 5 hours, 3 - 5 to 10 hours, or 4 - >10 hours)
failures	number of past class failures (numeric: n if 1<=n<3, else 4)
schoolsup	extra educational support (binary: yes or no)
famsup	family educational support (binary: yes or no)
paid	extra paid classes within the course subject(binary: yes or no)
activities	extra-curricular activities (binary: yes or no)
nursery	attended nursery school (binary: yes or no)
higher	wants to take higher education (binary: yes or no)
internet	Internet access at home (binary: yes or no)
romantic	with a romantic relationship (binary: yes or no)
famrel	quality of family relationships (numeric: from 1 - very bad to 5 - excellent)
freetime	free time after school (numeric: from 1 - very low to 5 - very high)
goout	going out with friends (numeric: from 1 - very low to 5 - very high)
Dalc	workday alcohol consumption (numeric: from 1 - very low to 5 - very high)
Walc	weekend alcohol consumption (numeric: from 1 - very low to 5 - very high)
health	current health status (numeric: from 1 - very bad to 5 - very good)
absences	number of school absences (numeric: from 0 to 93)
G1	first period grade (numeric: from 0 to 20)
G2	second period grade (numeric: from 0 to 20)
G3	final grade (numeric: from 0 to 20, output target)

Table 6.1: Dataset description

6.1.1 Settings

In order to better understand the effects of the methodology we follow and to compare results obtained in different contexts we design two settings. Each one presents a different configuration of the dataset which causes different value of inequality. *Setting 1* (S1) composed by 649 observations which is the original dataset, and *setting 2* (S2) composed by 741 observations with higher initial inequality. To compose *S2* we take observations with low outcome (less than 10) and no need of extra educational support from original dataset and multiply them in order to force a different distribution and increase inequality.

6.2 Experiment

6.2.1 The Experiment: first setting

Estimation of Types/Circumstances

As we can see from Figure 6.1 the computation of the types based on the circumstances and conditional inferred tree produced *7 different types.*

- **type A**: consists of individuals with no failures in past class, that want to take higher education, belonging to the school "GP" and who need extra educational support.
- **type B**: consists of individuals with no failures in past class, that want to take higher education, belonging to the school "GP", who do not need extra educational support and study more than 2 hours.
- **type C**: consists of individuals who have no failures in past class, want to take higher education, belonging to the school "GP", do not need extra educational support, study less than 2 hours and have father with secondary or higher education .
- **type D**: consists of individuals with 1 or more failures in past class.
- **type E**: consists of individuals who have no failures in past class, want to take higher education, belonging to the school "GP", do not need extra educational support, study less than 2 hours and have father with education between 5th and 9th grade, or primary education, or none.
- **type F**: consists of individuals who have no failures in past class, do not want to take higher education.
- **type G**: consists of individuals with no failures in past class, that want to take higher education, belonging to the school "MS".

Circumstances	Type
failures <= 0 & higher %in% c("no")	F
failures <= 0 & higher %in% c("yes") & school %in% c("GP") & schoolsup %in% c("no") & studytime <= 1 & Fedu <= 2	E
failures <= 0 & higher %in% c("yes") & school %in% c("GP") & schoolsup %in% c("no") & studytime <= 1 & Fedu > 2	C
failures <= 0 & higher %in% c("yes") & school %in% c("GP") & schoolsup %in% c("no") & studytime > 1	B
failures <= 0 & higher %in% c("yes") & school %in% c("GP") & schoolsup %in% c("yes")	A
failures <= 0 & higher %in% c("yes") & school %in% c("MS")	G
failures > 0	D

Figure 6.1: Circumstances explained for each type

In Figure 6.2 we have a summary of the types-distribution for each ranking.

Estimation of Effort

As expected, the cumulative distribution approximation through the Bernstein polynomial is better when we have larger partition, which happens when we have types highly populated. In Figure 6.3 we have a comparison between the *Empirical Cumulative Distribution Function (ECDF)* and Bernstein polynomial function for each type-specific outcome distribution.

Inequality

Figure 6.4 shows differences between estimated inequalities through the different ranking. *Gini before* column is based on the *G3* attribute which is the same value of the initial dataset. *Gini after* column is based on the *outcome* score which is valuated differently according to the specific ranking. **The highest levels of inequality is observed in the *need-500* ranking**. After the outcome redistribution the least level of inequality is achieved by the *equal-100* ranking. Observing the Δ *Gini* column we notice that **the highest variation, we could say improvement in equality, is obtained thanks to *equity-500* ranking**.

Diversity

Figure 6.5 shows that the most diverse attributes is our baseline score *G3* - which indicates the final grade of the student - which has the highest Shannon index. The least diverse one is *paid* which indicates if the student has taken extra paid lessons; only 57 students did it.

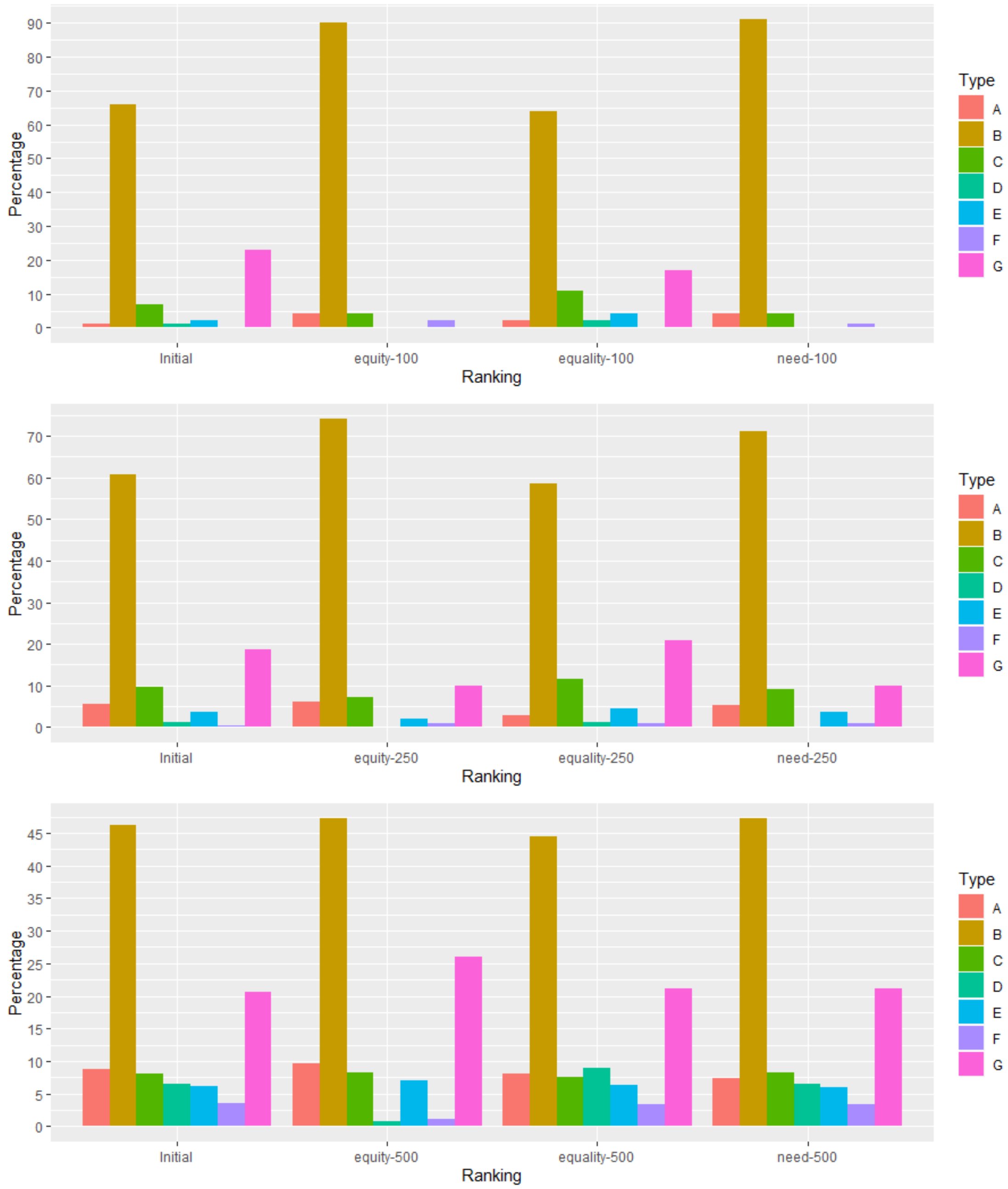

Figure 6.2: Types-distribution for each ranking

Entropy

As for diversity index, the entropy level measured in Figure 6.6 through the Theil index shows that ***need-500* presents the highest *G3* entropy**. **After**

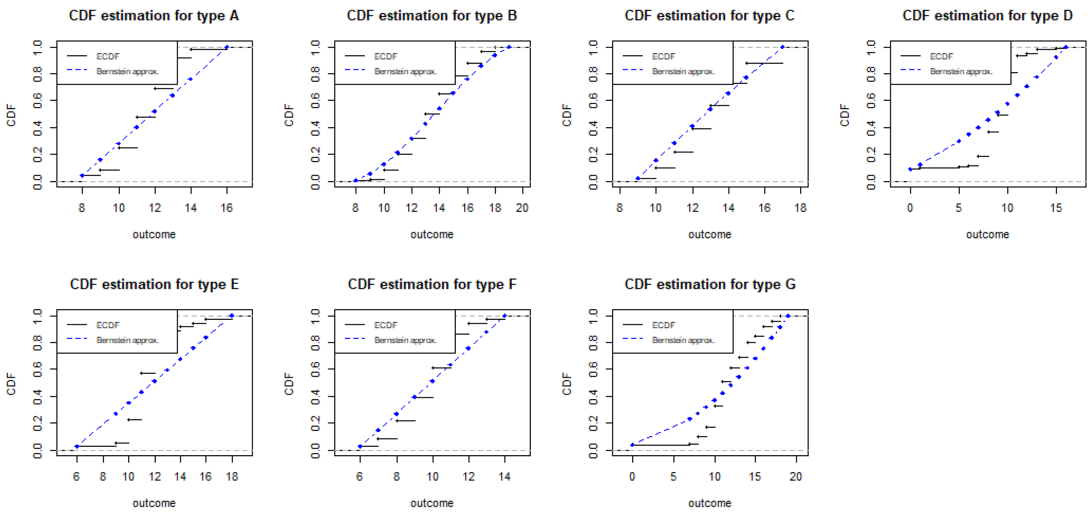

Figure 6.3: Comparison between the Empirical Cumulative Distribution Function (ECDF) and Bernstein polynomial function for each type-specific outcome distribution

Ranking	Gini before	Gini after	Δ Gini
equity-100	9.80%	6.50%	3.30%
equity-250	8.90%	5.20%	3.70%
equity-500	10.40%	6.30%	**4.10%**
equal-100	4.00%	3.50%	0.50%
equal-250	6.80%	4.10%	2.70%
equal-500	10.60%	**7.50%**	3.10%
need-100	9.70%	6.50%	3.20%
need-250	9.10%	5.40%	3.70%
need-500	**11.20%**	**7.50%**	3.70%

Figure 6.4: Gini index for G3 (before) and outcome (after) across different ranking

the outcome redistribution *need-500* still presents the highest *outcome* entropy, but in general all the entropy indexes result flattened so they are lower and more similar each other.

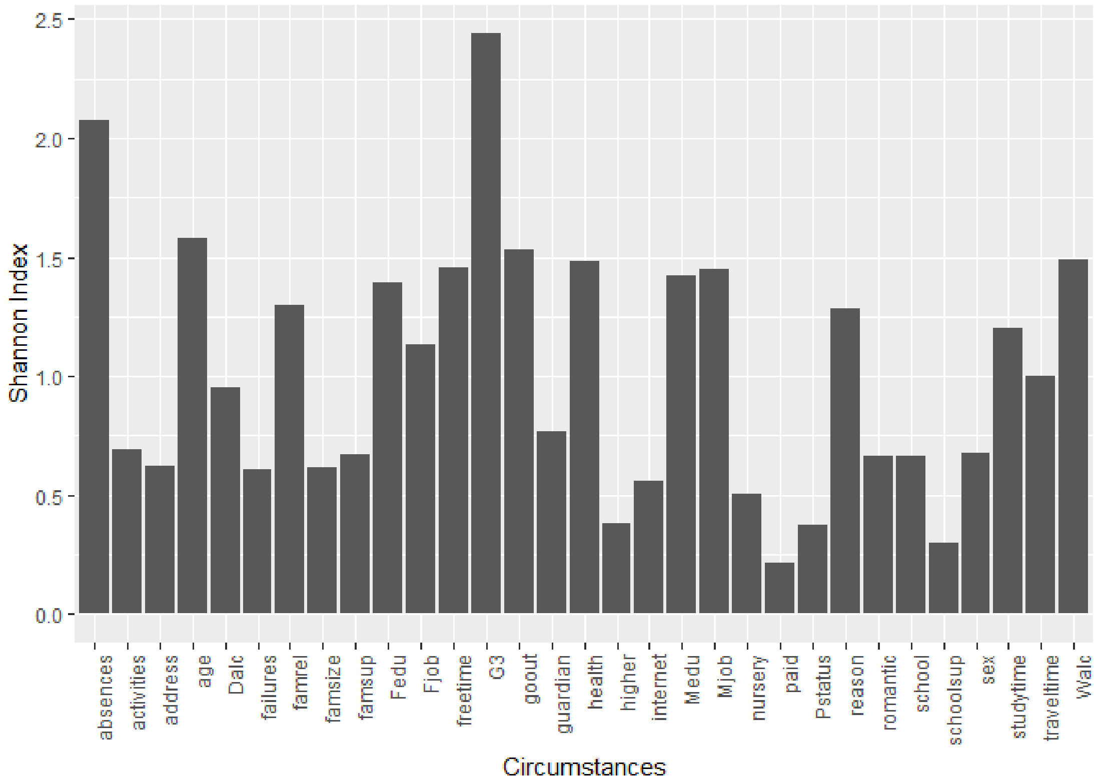

Figure 6.5: Shannon index for each attribute

Opportunity-Loss Profile

Figure 6.7 shows results in terms of *Opportunity Loss Profile*. First column indicates the ranking type in which the metric is evaluated. Following columns present the *type identifier* of the most advantaged and disadvantaged types. Columns marked with *"after"* contain results evaluated after the outcome redistribution. Differences of type are highlighted with green color. The *type F* is most disadvantaged across all ranking. After the redistribution in all rankings, dependently on the ranking size *type F,E,D* are classified as disadvantaged respectively for size *100,250,500* . *type B* in normal conditions is classified more often as advantaged. After the redistribution *type B* together with *type C* become the advantaged ones.

Opportunity-Loss Rate

In Figure 6.8 we can see how disadvantaged types get enhancement of their condition.

- The disadvantaged *type F* get the highest increase in *equity-250, equity-500,*

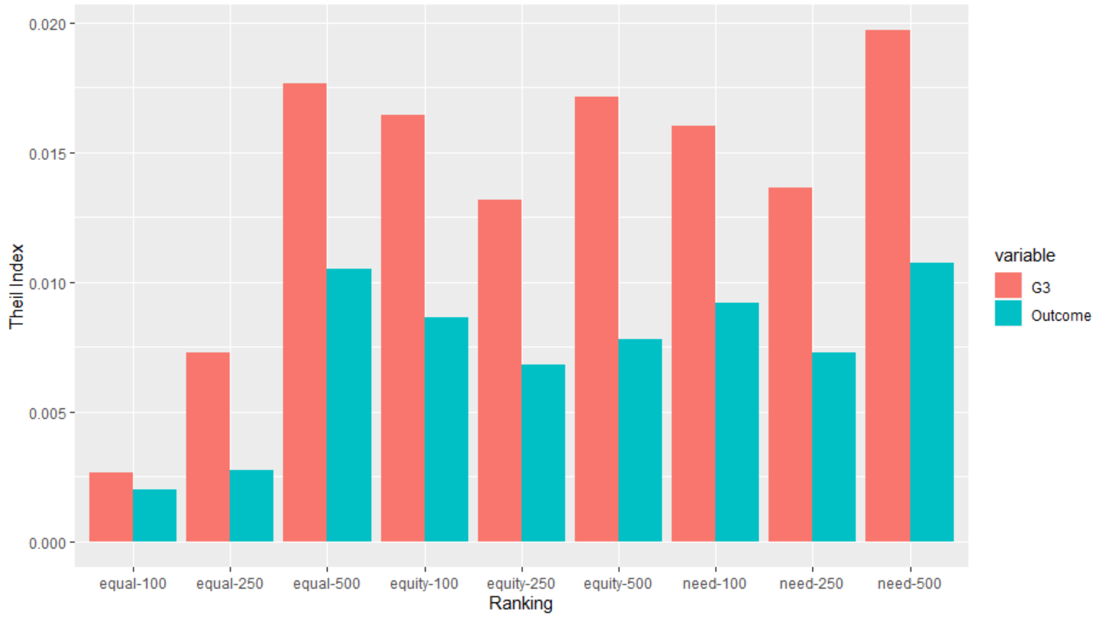

Figure 6.6: Theil index for G3 and outcome across different ranking

Ranking	Disadvantaged	Disadvantaged (after)	Advantaged	Advantaged (after)
equity-100	F	F	B	**C**
equity-250	F	**E**	G	**C**
equity-500	F	**D**	D	**B**
need-100	F	F	B	**C**
need-250	F	**E**	G	**B**
need-500	F	**D**	B	B

Figure 6.7: Opportunity Loss Profile across different ranking - some types report *NaN* due to their absence in that specific ranking

need-250. In those rankings, it is no longer the disadvantaged one after the redistribution.

- The advantaged *type B* get the highest decrease in *equity-100, and need-100*. In those rankings, it is no longer the advantaged one after the redistribution
- The *type D* get the highest decrease in *equity-500*. It becomes the disadvantaged one after the redistribution.

- The *type C* get the highest increase in *equity-100, and need-100.* In those rankings, it becomes the advantaged one after the redistribution.

Type	OLR_equity-100	OLR_equity-250	OLR_equity-500	OLR_need-100	OLR_need-250	OLR_need-500
A	0.640	0.211	0.047	0.646	0.244	1.000
B	-1.000	-0.368	-0.126	-1.000	-0.412	-0.080
C	1.000	-0.072	-0.211	1.000	-0.317	-0.300
F	0.207	1.000	1.000	0.220	1.000	0.578
E	NaN	-0.614	-0.095	NaN	-0.857	-0.074
G	NaN	-1.000	-0.369	NaN	-1.000	-1.000
D	NaN	NaN	-1.000	NaN	NaN	-0.373

Figure 6.8: Opportunity Loss Rate across different ranking

Reward Profile

Figure 6.9 presents a comparison of reward profile across different ranking. *Equal* rankings tend to perform more rewards in their less-populated version. Instead, *equity* rankings perform more rewards in their more-populated version. Furthermore, *equity* policy is the one who performs the highest number of rewarding. *Type D* is the one most frequently penalized. *Type B and C* are most frequently positively rewarded.

Reward Rate

Figure 6.10 presents a comparison across different ranking of reward rate. *Equity-500 and equal-100* mostly penalize *type D* which becomes the most penalized after the policy application. *Equal-100* mostly positively reward *type B* which becomes the most rewarded after the policy application.

Distributive Rate

Figure 6.11 presents a comparison across different ranking of mean outcome and mean distributive rate calculated inside each ranking. Ranking based on *equality* have the lowest mean outcome, and it is almost constant across different ranking sizes. *Equity and need* ranking have comparable mean outcome, which tends to

Ranking	Disadvantaged	Disadvantaged (after)	Advantaged	Advantaged (after)
equity-100	F	F	B	**C**
equity-250	F	**E**	G	**C**
equity-500	F	**D**	D	**B**
equal-100	A	**D**	G	**B**
equal-250	F	F	G	**B**
equal-500	D	D	B	B
need-100	F	F	B	**C**
need-250	F	**E**	G	**A**
need-500	F	**D**	B	B

Figure 6.9: Reward Profile

Type	RR_equity-100	RR_equity-250	RR_equity-500	RR_equal-100	RR_equal-250	RR_equal-500	RR_need-100	RR_need-250	RR_need-500
A	1.000	0.337	0.101	0.985	0.226	1.000	1.000	0.390	1.000
B	-1.000	-0.361	-0.122	1.000	0.647	0.054	-1.000	-0.406	-0.140
C	0.835	-0.070	-0.206	0.828	0.561	-0.307	0.838	-0.310	-0.340
F	0.118	1.000	1.000	NaN	-0.437	-0.407	0.131	1.000	0.499
E	NaN	-0.604	-0.069	0.658	1.000	0.318	NaN	-0.846	-0.096
G	NaN	-1.000	-0.371	-0.712	-0.081	-1.000	NaN	-1.000	-1.000
D	NaN	NaN	-1.000	-1.000	-1.000	-0.497	NaN	NaN	-0.416

Figure 6.10: Reward Rate

decrease when ranking size increases. The highest mean outcome improvement is in *equal-100*. All the mean distributive rate are positive, which means that on average the individual increase their outcome following the redistribution.

6.2.2 The Experiment: second setting

Estimation of Types/Circumstances

In Figure 6.12 we have a summary of the types-distribution for each ranking. Clearly types and circumstances remain the same of *S1*.

Ranking	Mean Outcome	Mean Distributive Rate
equal-100	12.570	33.76%
equal-250	12.824	0.95%
equal-500	12.543	29.04%
equity-100	15.357	9.42%
equity-250	14.037	9.19%
equity-500	12.936	24.84%
need-100	15.355	10.84%
need-250	14.009	12.25%
need-500	12.774	25.06%

Figure 6.11: Mean outcome and mean Distributive rate for each ranking

Estimation of Effort

Also here the cumulative distribution approximation through the Bernstein polynomial is better when we have larger partition, which happens when we have types highly populated. In Figure 6.13 we have a comparison between the *Empirical Cumulative Distribution Function (ECDF)* and Bernstein polynomial function for each type-specific outcome distribution.

Inequality

Figure 6.14 shows differences between estimated inequalities through the different ranking. **The highest levels of inequality are observed in the *need-500* ranking**. After the outcome redistribution the least level of inequality is still belonging to the *equal-100* ranking, even if it is subject to a slight increment of 0.80%. The ranking with the highest inequality become *equal-500*, but still in *need-500* it remains very high (8.70%). Observing the Δ *Gini* we notice that **the highest variation, we could say improvement in equality, is obtained in *equity-500* ranking**, with *equity-250* and *need-500* tied for second position.

Diversity

Figure 6.15 shows comparison of diversity between setting 1 (named *original* and setting 2 named *modified*). Results are comparable.

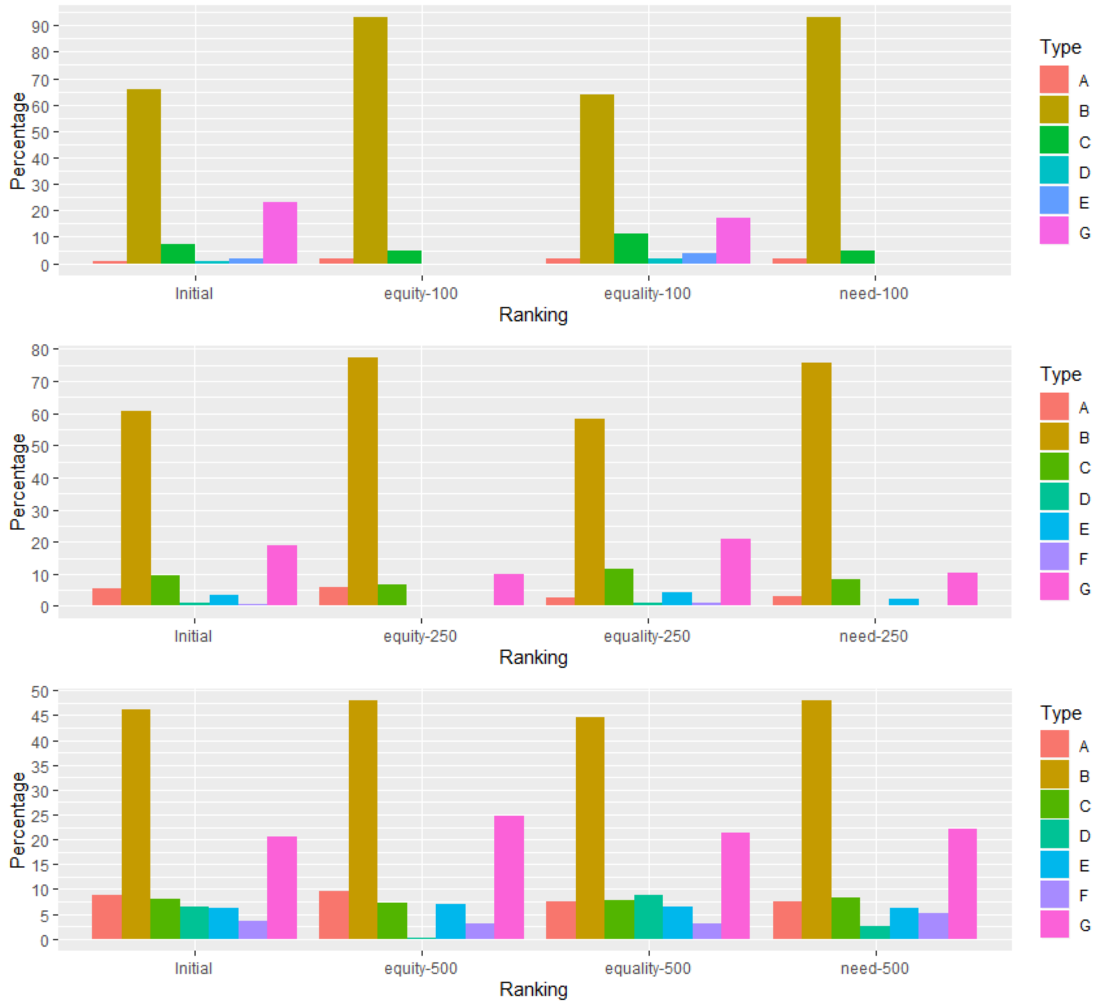

Figure 6.12: Types-distribution for each ranking on 2^{nd} setting

Entropy

As for diversity index, the entropy level measured in Figure 6.16 through the Theil index shows that ***need-500* present the highest *G3* entropy**. **After the outcome redistribution all but *equal-100* ranking are subject to entropy reduction** so they are lower and more similar each other.

Opportunity-Loss Profile

Figure 6.17 shows *Opportunity Loss Profile*.The *type A* is the most often disadvantaged across all ranking except for all the ranking versions with size 500. After the redistribution it remains most disadvantaged, even if additional types appear as

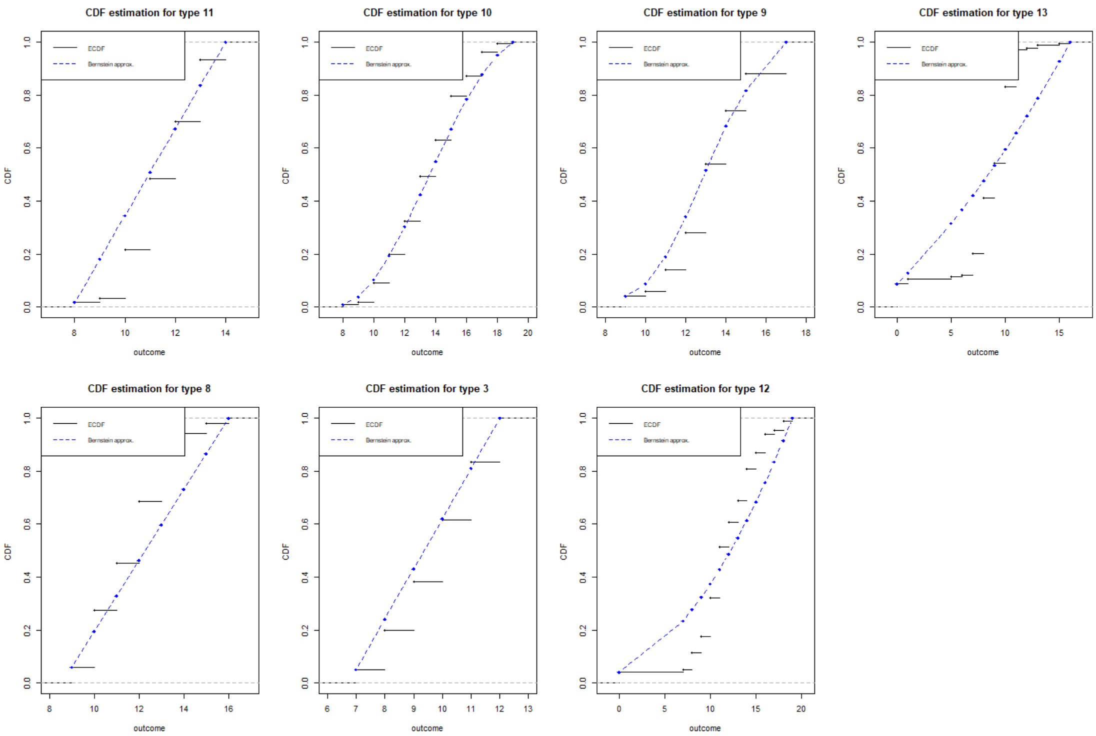

Figure 6.13: Comparison between the Empirical Cumulative Distribution Function (ECDF) and Bernstein polynomial function for each type-specific outcome distribution on 2^{nd} setting

Ranking	Gini before	Gini after	Δ Gini
equity-100	8.20%	6.30%	1.90%
equity-250	9.20%	6.10%	3.10%
equity-500	11.10%	7.80%	**3.30%**
equal-100	4.00%	4.80%	**-0.80%**
equal-250	6.80%	5.20%	1.60%
equal-500	10.40%	**9.00%**	1.40%
need-100	8.20%	6.30%	1.90%
need-250	8.90%	6.40%	2.50%
need-500	**11.80%**	8.70%	3.10%

Figure 6.14: Gini index for G3 (before) and outcome (after) across different ranking on 2^{nd} setting

disadvantaged in the list. In normal conditions *type B* represents the advantaged one but with *type G and D* who also appear in the list. After the redistribution situation is more clear and *type B* is still the most advantaged one.

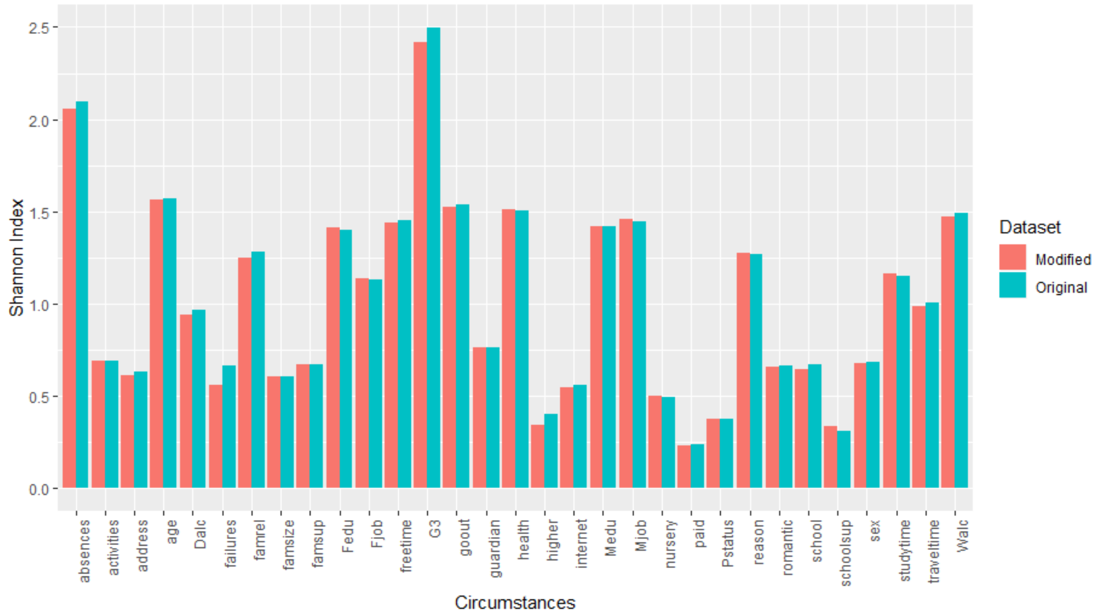

Figure 6.15: Comparison of Shannon index for each attribute

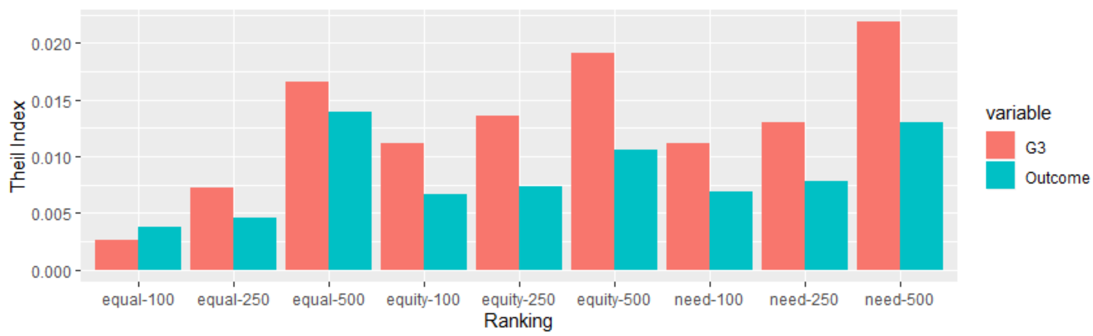

Figure 6.16: Theil index for G3 and outcome across different ranking on 2^{nd} setting

Opportunity-Loss Rate

In Figure 6.18 we can see - for each type - average shifts of outcome values.

- The disadvantaged *type F* get the highest increase in *equity-500, need-500.* In *need-500* it is no longer the disadvantaged one after the redistribution.
- The most disadvantaged *type A* get the highest increase in *equity-250, need-250.* In *need-250* it is no longer the disadvantaged one after the redistribution.
- The advantaged *type B* get the highest decrease in *equity-100, and need-100.*

Ranking	Disadvantaged	Disadvantaged (after)	Advantaged	Advantaged (after)
equity-100	A	A	B	**C**
equity-250	A	A	G	**B**
equity-500	F	F	D	**B**
need-100	A	A	B	**C**
need-250	A	**E**	G	**B**
need-500	F	**D**	B	B

Figure 6.17: Opportunity Loss Profile across different ranking on 2^{nd} setting

In those rankings, it is no longer the advantaged one after the redistribution

- The *type D* get the highest decrease in *equity-500, need-500.* In *equity-500* it is no longer the advantaged one after the redistribution.
- The *type C* get the highest increase in *equity-100, and need-100.* In those rankings, it becomes the advantaged one after the redistribution.

Type	OLR_equity-100	OLR_equity-250	OLR_equity-500	OLR_need-100	OLR_need-250	OLR_need-500
A	0.687	1.000	0.522	0.687	1.000	0.974
B	-1.000	0.488	0.557	-1.000	0.289	0.414
C	1.000	0.870	0.410	1.000	0.294	-0.637
G	NaN	-1.000	0.161	NaN	-0.981	-0.969
D	NaN	NaN	-1.000	NaN	NaN	-1.000
E	NaN	NaN	0.321	NaN	-1.000	-0.020
F	NaN	NaN	1.000	NaN	NaN	1.000

Figure 6.18: Opportunity Loss Rate across different ranking on 2^{nd} setting

Reward Profile

Figure 6.19 presents a comparison of reward profile across different ranking. *Equal* rankings tend to perform more rewards in their less-populated version. Instead, *equity* rankings perform more rewards in their more-populated version. Furthermore, *equity* policy is the one who performs the highest number of rewarding. *Type D* is the one most frequently penalized. *Type B and C* are most frequently positively rewarded.

Ranking	Disadvantaged	Disadvantaged (after)	Advantaged	Advantaged (after)
equity-100	A	A	B	**C**
equity-250	A	A	G	**B**
equity-500	F	**D**	D	**B**
equal-100	A	**D**	G	**B**
equal-250	F	F	G	**B**
equal-500	D	D	B	B
need-100	A	A	B	**C**
need-250	A	**E**	G	**B**
need-500	F	**D**	B	B

Figure 6.19: Reward Profile on 2^{nd} setting

Reward Rate

Figure 6.20 presents a comparison across different ranking of reward rate. *Equity-500 and equal-100* mostly penalize *type D* which becomes the most penalized after the policy application. *Equal-100 and equal-250* mostly positively reward *type B* which becomes the most rewarded after the policy application. *Need-100 and equity-100* mostly positively reward *type C* which becomes the most rewarded after the policy application.

Type	RR_equity-100	RR_equity-250	RR_equity-500	RR_equal-100	RR_equal-250	RR_equal-500	RR_need-100	RR_need-250	RR_need-500
A	0.548	1.000	0.527	0.976	0.630	0.802	0.548	1.000	0.963
B	-1.000	0.500	0.560	1.000	1.000	1.000	-1.000	0.303	0.391
C	1.000	0.864	0.422	-0.837	0.554	-0.363	1.000	0.306	-0.621
G	NaN	-1.000	0.165	-0.636	0.341	-0.633	NaN	-1.000	-1.000
D	NaN	NaN	-1.000	-1.000	-1.000	-1.000	NaN	NaN	-0.992
E	NaN	NaN	0.337	0.679	0.982	0.349	NaN	-0.991	-0.022
F	NaN	NaN	1.000	NaN	-0.003	-0.101	NaN	NaN	1.000

Figure 6.20: Reward Rate on 2^{nd} setting

Distributive Rate

Figure 6.21 presents a comparison across different ranking of mean outcome and mean distributive rate calculated inside each ranking. Ranking based on *equality* have the lowest mean outcome, and it is almost constant across different ranking sizes. *Equity and need* ranking have comparable mean outcome, which tends to decrease when ranking size increases. The highest mean outcome improvement (39.53%) is in *equal-250*. Unlike any other, in *equity-250*, *need-100*, and *need-250* the mean distributive rate is negative, which means that on average the individual lost their outcome following the redistribution.

Ranking	Mean Outcome	Mean Distributive Rate
equal-100	12.290	33.81%
equal-250	12.705	39.53%
equal-500	12.513	16.71%
equity-100	15.960	2.63%
equity-250	14.344	-5.84%
equity-500	12.958	12.93%
need-100	15.959	2.64%
need-250	14.304	-5.22%
need-500	12.837	12.63%

Figure 6.21: Mean outcome and mean Distributive rate for each ranking on 2^{nd} setting

6.3 Results and discussion

We evaluate the results of the selection process varying the number of available seats. We compare the top students selected in case of 100, 250, and 500 free places. We repeated the same analysis with two settings and we following analyze the differences:

Types-distribution: we notice in size-100 ranking *type F* completely disappears in *S2* despite there are more observations. Furthermore *type B* is slightly more present. In *need-250 and equity-250* of *S2* there are less types present. Both *S2 and S1* lose completely *type D* in *need-500 and equity-500*. *S1* lose *type A* as well.

Diversity: as we can see from 6.15 the Shannon index is almost the same. There is no relevant change in diversity.

Inequality: *need-500* presents the highest initial inequalities both in *S1 and S2*. After the redistribution *equal-500* presents the highest initial inequalities both in *S1 and S2*. In *equality-500* we can observe the highest improvement in terms of inequality (it decreases) both in *S1 and S2*, however percentage is higher in *S1*. In general *S1* presents the highest improvement across all ranking, instead in *S2* we actually have a small worsening. In *S1* the percentage of improvement grows with ranking size. The same trend is present in *S2* except for *equal* ranking.

Opportunity-Loss Profile: results here are very similar. The main difference is given by the initial disadvantaged types, which are *type F and type A* respectively for *S1 and S2*.

Opportunity-Loss Rate: *equity-100* of *S2* presents higher variation for the advantaged *type C*. Almost all types of *S1* are subject to decrease in *equity-250 and equity-500* while in *S2* they positively increase.

Distributive Rate: mean outcome is slightly lower in *S2*, but it is quite similar in *S1*. Positive variations in *equal-500* of *S2* become negative in *S1*. The highest distributive mean rate in *equity-250* of *S1* become a small decrease in *S2*. Finally we have a decrease in *need-250* in both settings *S1 and S2*.

In general most of the metric used show similar behaviour in both settings *S1 and S2*.

- Better performances in terms of inequality reduction are obtained thanks to *equity* ranking. *Equity* rankings not only have the highest percentage of inequality decrease, but also have the highest *power* of outcome redistribution. Considering its design, with the policy's application the initial outcome *G3* is subject to numerous mutations which tends to flatten the outcome and therefore to reduce inequalities. It is worth noting that *equity* and *equality* policies are not compatible. In fact, *need* ranking, which should benefit from the combined approach, actually has lower performance than *equity* in terms of inequality reduction.

- The outcome redistribution generally causes an entropy decrease in every ranking except for *equal-100* of *S2.Equal* ranking in its smallest version, due to their constraints on keeping equal the number of men and women, are less effective than other in reducing entropy. However it represents an isolated case, and therefore may not be significant.

- The *equality* ranking are populated with higher variety of population types compared to *equity and need* who often sacrifice some types. However, the inter groups' equality constraint leads *equality* rankings to the lowest mean outcome, which remains constant when ranking size increases. *Equity* rankings have the highest mean outcome in all settings and sizes. Considering the meaning of the estimated outcome, which is the real performance value that individuals would have had if they were born in the same circumstances, using *equity* rankings leads to more benefits for all the stakeholders. The more they are fair, the more they *get* valuables candidates.

- . The mean distributive rate is pretty variable and sometimes even negative. It give us unforeseen results, especially in *S2* where we would have expected the greatest mean redistribution. It certainly deserves further analysis.

Chapter 7

Conclusions

Algorithmic tools are likely to be used more and more in every process affecting our ordinary people's everyday life. Analysis on how these systems are being used often led to evidence of different impact of the decisions issued by such systems on different groups of population, causing discrimination. In addition, lack of transparency and accountability caused the developing companies to be criticized and exposed to further investigation.

Discriminating behaviors may arise for many reasons, thus before deployment of such tools it is necessary a risks assessment and evaluation of their impact on our society. It is in fact widely recognized that ADS/RS results may have a crucial influence on people career and business opportunities, educational placement, access to benefits, and even social and reproductive success [12]. It is therefore of societal and ethical importance to investigate whether such algorithms provide outcomes that can declass, demerit, or exclude individuals of disadvantaged groups (e.g., racial or gender discrimination)[13].

We give our contribution in the field of algorithmic fairness experimenting the effects of distributive justice criteria applied to the world of ranking systems. In particular we combine methods from the philosophical, sociological, economic sciences and machine learning in order to develop practical ranking algorithms based on *equity, equality, and need*. We have seen how in some situations these algorithms, *equity* ranking in particular, may produce positive effects on the population and reduce social inequalities. These results, which still need further improvements, are promising and proved the effectiveness on ranking systems area of methods which do not belong to computer science, hence open up new avenues for the multidisciplinary research. Future works may include the application of developed ranking algorithms on different datasets, testing performances with different sensitive attributes such as ethnicity and different scenarios such as job recruiting.

Appendix A

The Algorithm (code)

```
quantili <- seq(from=0.2,to=1.0,by=0.2)
outcome <- "G3"
customGreen0 = "#DeF7E9"

customGreen = "#71CA97"

customRed = "#ff7f7f"

sign_formatter <- formatter("span",
                              style = x ~ style(color = ifelse(x > 0, "green",
                                                                  ifelse(x < 0, "red", "black"))))
unit.scale <- function(x) (x - min(x)) / (max(x) - min(x))

colorbar <- function(color = "lightgray", fun = "comma", digits = 0) {
  fun <- match.fun(fun)
  formatter("span", x ~ fun(x, digits = digits),
            style = function(y) style(
              display = "inline-block",
              direction = ifelse(y > 0, "rtl","ltr") ,
              "border-radius" = "4px",
              "padding-right" = "2px",
              "background-color" = ifelse(y > 0, csscolor(color), customRed) ,
              width = percent(proportion(as.numeric(y))),
              "font-weight" = ifelse(y == max(y), "bold", NA)
            )
  )
}
```

```r

mapptype <- function(x){
 LETTERS[match(x,tipi)]
}

brunori_bernstein <- function(hr){

  tipi = unique(hr$node_placement)

  ### creo un dataframe per ospitare i gradi trovati relativi ai tipi
  grado <- c(1:length(tipi))*0
  info = data.frame(tipi,grado)

  for(k in tipi) {

    ### seleziono il subset del tipo k
    y = hr[hr$node_placement == k,]

    # definisco il numero di fold
    f_max = 10
    # creo i fold in modo che siano bilanciati per la variabile di factor
    folds <- createFolds(factor(y[[outcome]]), k = f_max, list = FALSE)
    # assegno ad ogni riga il valore del rispettivo fold di appartenenza
    y$fold = folds

    # scelgo un range di gradi del polinomio tra provare
    range_b = 1:10
    # faccio un vettore per ospitare le rispettive likelihood
    LLs <- c(range_b)*0

    ### b è il grado del polinomio approssimatore di bernstein
    for(b in range_b){

      range_f = 1:f_max
      # faccio un vettore che ospita le likelihood per ogni grado
      LLsb <- c(range_f)*0

      ordine <- b

      for(f in range_f){

        trainData = y[y$fold != f,]
        trainData <- trainData[[outcome]]
        testData = y[y$fold == f,]
        testData <-testData[[outcome]]

```

```
        eCDF <- ecdf(trainData)

        m <- min(trainData)
        M <- max(trainData)

        m_test <- min(testData)
        M_test <- max(testData)

        # riporta il dominio della eCDF del train tra 0 e 1
        Fy <- function(y){
          b <- M
          a <- m
          eCDF((M-m)*y+m)
        }

        ### stimo i coefficienti di bernstein approssimando la CDF del training set
        bc = bernstein(Fy, dims = 1, k = ordine)
        ### coefficienti estratti dal polinomio di bernstein calcolato sul training set
        cf <- bc$coeffs

        ### set-up basis - il primo argomento dev'essere un tipo 'numeric_var'
        bb <- Bernstein_basis(numeric_var("x", support = c(m_test, M_test)),
                              order = ordine, ui = "increasing")

        x <- sort(testData)
        xx <- as.data.frame(x)
        LLsb[f] = sum(log(predict(bb, newdata = xx, coef = cf, deriv = c(x = 1))))
      }
      LLs[b] = sum(LLsb)
    }

    ### massima likelihood calcolata
    max_b = max(LLs)
    ### indice di LLs dove si trova la massima likelihood
    grado_m = match(max_b,LLs)
    ### assegna quell'indice al tipo relativo - è il grado del polinomio
    info[info$tipi == k,]$grado = grado_m
  }
  return (info)
}


giveme_x <- function(z) {
```

```
  x <- sort(z)
  xx <- as.data.frame(x)
  return (xx)
}

normalize_var <- function(array, x, y){
  # Normalize to [0, 1]:
  m = min(array)
  range = max(array) - m
  array = (array - m) / range

  # Then scale to [x,y]:
  range2 = y - x
  normalized = (array*range2) + x
  return (round(normalized,3))
}

### restituisce la CDF della distribuzione x stimata secondo
    Bernstein con il grado m
bern_app <- function(x,m){

  # distribuzione di cui stimare la Bernstein(CDF)
  z <- x

  Fz <- ecdf(z)

  # funzione tra 0 e 1 da dare in input a bernstein
  f <- function(y){
    ### FF funzione da calcolare
    ### z dominio di FF
    ### y punto/i dove calcolare FF
    FF <- Fz
    z <- z

    b <- max(z)
    a <- min(z)
    FF((b-a)*y+a)
  }

  cf <- bernstein(f, dims = 1, k = m)$coeffs
  bb <- Bernstein_basis(numeric_var("x", support = c(min(z), max(z)))
    ,
                        order = m, ui = "increasing")

  newList <- list("basis" = bb, "cf" = cf)

  return (newList)
}

```

```
bern_app2 <- function(x,m){

  # distribuzione di cui stimare la Bernstein(CDF)
  z <- x

  Fz <- ecdf(z)

  # funzione tra 0 e 1 da dare in input a bernstein
  f <- function(y){
    ### FF funzione da calcolare
    ### z dominio di FF
    ### y punto/i dove calcolare FF
    FF <- Fz
    z <- z

    b <- max(z)
    a <- min(z)
    FF((b-a)*y+a)
  }

  cf <- bernstein(f, dims = 1, k = m)$coeffs
  bb <- Bernstein_basis(numeric_var("x", support = c(min(z), max(z)))
    ,
                        order = m, ui = "increasing")

  x <- sort(z)
  xx <- as.data.frame(x)

  predict(bb, newdata = xx, coef = cf)
}

brunori_outcome <- function(df, info){
  hr <- df
  hr$quantile <- 0
  tipi = info$tipi

  for(k in tipi) {
    ### seleziono il subset del tipo k
    y <- hr[hr$node_placement == k,]
    m <- info[info$tipi == k,]$grado

    da <- function(x){
      yy <- y[[outcome]]
      ba <- bern_app(yy, m)
      round(predict(ba$basis, newdata = givеme_x(x), coef = ba$cf),
    digits = 2)
    }

    ecdfs <- c(1:length(y[[outcome]]))*0
```

```
    for (i in 1:length(y[[outcome]])){
      yy <- y[[outcome]]
      a <- da(yy[i])
      ecdfs[i] <- a
    }
    y$cdf <- ecdfs

    last_quant <- -1
    h <- 1
    for(quantile in quantili){

      hr[ rownames(y[(y$cdf > last_quant) & (y$cdf <= quantile),]),]$
    quantile <- h
      last_quant <- quantile
      h <- h+1
    }

  }

  # PER OGNI TIPO
  # per ogni quantile
  # selezionare outcome (Age)
  # calcolare la media di outcome per tutta la popolazione
  # calcolare all'interno del quantile la media dell'outcome

  mu <- mean(hr[[outcome]])
  media_k_q <-c(1:length(tipi))*0
  medie_q <-c(1:length(quantili))*0

  for(quantile in 1:5) {
    hh <- hr[hr$quantile == quantile,]
    medie_q[quantile] <- mean(hh[[outcome]])
  }

  hr$outcome <- 0

  for(k in tipi){
    for (quantile in 1:5){
      hh <- hr[(hr$node_placement == k) & (hr$quantile == quantile)
    ,]
      y <- hh[[outcome]]
      hr[(hr$node_placement == k) & (hr$quantile == quantile),]$
    outcome <- y* (mu/medie_q[quantile])
    }
  }
  return(hr)
}

opportunity_lossprofile <- function(df){
```

```
 
 hr <- df
 tipi = unique(hr$node_placement)
### avremo un tipo maggiormente deprivato per ogni quantile
 deprivati <- c(1:length(quantili))*0
 deprivati_std <- c(1:length(quantili))*0
 fortunati <- c(1:length(quantili))*0
 fortunati_std <- c(1:length(quantili))*0

 for(quantile in 1:length(quantili)){

   ### calcoliamo un outcome medio per ogni tipo
   outcomes <- c(1:length(tipi))*0
   outcomes_std <- c(1:length(tipi))*0

   for (k in tipi){
     ### inseriamo il valore medio alla posizione corrispondente
     ### NB: l'indice del valore inserito in outcomes == indice di k in "tipi"
     outcomes_std[which(k == tipi)] <- mean(hr[which(hr$node_placement == k),]$outcome)

     d <- hr[which(hr$node_placement == k),]
     outcomes[which(k == tipi)] <- mean(d[[outcome]])

   }
   ### dagli outcome calcolati estraiamo il minore e ricaviamo il suo indice
   indicetipominore <- match(min(outcomes),outcomes)
   indicetipominore_std <- match(min(outcomes_std),outcomes_std)

   indicetipomaggiore <- match(max(outcomes),outcomes)
   indicetipomaggiore_std <- match(max(outcomes_std),outcomes_std)

   ### questo indice è uguale a quello del tipo da cui proviene rispetto a "tipi"
   tipo1 <- tipi[indicetipominore]
   tipo2 <- tipi[indicetipominore_std]
   tipo3 <- tipi[indicetipomaggiore]
   tipo4 <- tipi[indicetipomaggiore_std]

   deprivati[quantile] <- tipo1
   deprivati_std[quantile] <- tipo2
   fortunati[quantile] <- tipo3
   fortunati_std[quantile] <- tipo4
 }

 depr <- data.frame(table(deprivati))
 depr <- depr[order(depr$Freq),]
```

```

  depr_std <- data.frame(table(deprivati_std))
  depr_std <- depr_std[order(depr_std$Freq),]

  fort <- data.frame(table(fortunati))
  fort <- fort[order(fort$Freq),]

  fort_std <- data.frame(table(fortunati_std))
  fort_std <- fort_std[order(fort_std$Freq),]

  newList <- list("depr" = depr, "depr_std" = depr_std, "fort"=fort,
    "fort_std"=fort_std)

  return (newList)
}

stampa_olp <- function(rk_list) {

  ranking <- names(rk_list)
  deprived <- c(1:length(rk_list))
  deprived_std <- c(1:length(rk_list))
  privileged <- c(1:length(rk_list))
  privileged_std <- c(1:length(rk_list))
  h <- 1

  for(rk in rk_list) {
    o <- olp_table(opportunity_lossprofile(rk))
    deprived[h] <- o$Deprivati
    deprived_std[h] <- o$Deprivati_standard
    privileged[h] <- o$Privilegiati
    privileged_std[h] <- o$Privilegiati_standard
    h <- h + 1
  }
  info <- data.frame(ranking,deprived,deprived_std,privileged,
    privileged_std,stringsAsFactors=FALSE)
  names(info) <- c("Ranking","Disadvantaged", "Disadvantaged (after)"
    , "Advantaged", "Advantaged (after)")
  formattable(info,
              align =c("l","r", "l","r","l"),
              list(
                'Disadvantaged (after)' = formatter(
                  "span",
                  style = ~style(color= ifelse( `Disadvantaged` == `
    Disadvantaged (after)`, "black","green"),
                                  "font-weight"= ifelse( `
    Disadvantaged` == `Disadvantaged (after)`, NA,"bold"))),

                'Advantaged (after)' = formatter(
                  "span",
```

```
                    style = ~style(color= ifelse( `Advantaged` == `Advantaged (after)`, "black","green"),
                                "font-weight"= ifelse( `Advantaged` == `Advantaged (after)`, NA,"bold")))
                  )
                )

}

calcolo_ineq <- function(df,name) {
  ### CALCOLO INEQ
  #print(ineq(df$outcome, type="Gini"))

  plot(Lc(df$outcome),col="orange",lwd=2,main = paste0("Lorent Curve for ",name))
  #print(ineq(df[[outcome]], type="Gini"))
  #par(new = TRUE)
  lines(Lc(df[[outcome]]),col="blue",lwd=2, lty = 2)
  legend("topleft", legend=c("Outcome std", "Outcome before"),
         col=c("orange", "blue"), lty=1:2,cex = 0.7, horiz = TRUE)
}

stampa_ineq <- function(rk_list) {
  ranking <- names(rk_list)
  Gini_before <- c(1:length(rk_list))
  Gini_after <- c(1:length(rk_list))
  h <- 1
  par(mfrow=c(3,2))
  for(rk in rk_list) {
    calcolo_ineq(rk, ranking[h])
    Gini_after[h] <- round(ineq(rk$outcome, type="Gini"),3)
    Gini_before[h] <- round(ineq(rk[[outcome]], type="Gini"),3)
    h <- h + 1
  }

  info <- data.frame(ranking,Gini_before,Gini_after)
  info$delta_gini <- percent(info[["Gini_before"]] - info[["Gini_after"]])
  names(info) <- c("Ranking","Gini before", "Gini after", "&#916 Gini")

  formattable(info,
              align =c("l","r", "r","r"),
              list(
                'Gini before' = colorbar(color = "lightblue", fun = "percent", digits = 2),
                'Gini after' = colorbar(color = customGreen0, fun = "percent", digits = 2),
                '&#916 Gini' = formatter(
```

```
                    "span", x ~ percent(x),
                     style = ~style(color= ifelse( (`Gini after` - `
    Gini before`) < 0, "green","red"),

                                                      "font-weight" = ifelse((`Gini
    before` - `Gini after`) > 0,
                                                                              ifelse((`
    Gini before` - `Gini after`) == max(`Gini before` - `Gini after`),
     "bold",NA),
                                                                              ifelse((`
    Gini before` - `Gini after`) == min(`Gini before` - `Gini after`),
    "bold",NA)
                                                      )
                                 )
                             )

                             ))
}

stampa_olr <- function(rk_list) {
  ranking <- names(rk_list)

  olr_df <- lapply(rk_list, function(df){
    r <- opportunity_lossrate(df)
    r <- r[!duplicated(r$node_placement),]
    olr <- data.frame(r$node_placement, r$OpportunityLossRate)
    colnames(olr) <- c('Type','OpportunityLossRate')
    return (olr)
  })
  h <- 1
  for(name in ranking) {
    colnames(olr_df[[h]]) <- c('Type',paste0('OLR_',name))
    h <- h+ 1
  }
  rbl <- rbindlist(olr_df, fill = TRUE)
  rbl <- rbl %>%
    group_by(Type) %>%
    summarise_each(funs(mean(., na.rm = TRUE)))

  df <- rbl
  print(
    formattable(df,
      list(formattable::area(col = 2:(length(rk_list)+1)) ~ color_
    tile(customRed, "lightblue"),

             Type = formatter("span", style = ~ style(color = "black",
    font.weight = "bold")))
    )
  )
```

```r
}

metodo_brunori <- function(df) {
  info <- brunori_bernstein(df)
  df <- brunori_outcome(df,info)
  ml <- list("info" = info, "df" = df)
  return(ml)
}

ranking_top <- function(df, sortby) {
  ord_hr <- df[order(-df[[sortby]]),]

  top500 <- head(ord_hr,500)
  top250 <- head(ord_hr,250)
  top100<- head(ord_hr,100)

  newList <- list("top100" = top100, "top250" = top250, "top500"= top500)

  return (newList)
}

rank_equity <- function (df, k) {
  test4 <- df[order(-df$outcome),]

 num_groups = 50

  subsets <- test4 %>%
    group_by((row_number()-1) %/% (n()/num_groups)) %>%
    nest %>% pull(data)

  subsets <- lapply(subsets, function(set){
    setDT(set)[, mean_outcome_type := mean(outcome), by = node_placement]
  })

  l2<-lapply(subsets, function(x)
    cbind(x, outcome_1 = mean(x$mean_outcome_type)))

  equal <- rbindlist(l2)
  equal$mean_outcome_type <- NULL
  equal$outcome <- equal$outcome_1
  equal$outcome_1 <- NULL
  equal <- equal[order(-equal$outcome),]


  return(head(equal,k))
```

```
}

rank_equality <- function(df, column, k) {
  classi <- unique(df[[column]])
  len <- length(classi)
  size <- k/len
  df %>%
    arrange(desc(G3)) %>%
    group_by(sex) %>% slice(1:size)
}

rank_needing <- function(df, column, k) {
  classi <- unique(df[[column]])
  len <- length(classi)
  size <- k/len

  test4 <- df[order(-df$outcome),]

  num_groups = 50

  subsets <- test4 %>%
    group_by((row_number()-1) %/% (n()/num_groups)) %>%
    nest %>% pull(data)

  subsets <- lapply(subsets, function(set){
    setDT(set)[, mean_outcome_type := mean(outcome), by = node_
  placement]
  })

  l2<-lapply(subsets, function(x)
    cbind(x, outcome_1 = mean(x$mean_outcome_type)))

  equal <- rbindlist(l2)
  equal$mean_outcome_type <- NULL
  equal$outcome <- equal$outcome_1
  equal$outcome_1 <- NULL
  df <- equal[order(-equal$outcome),]

  df %>%
    arrange(desc(outcome)) %>%
    group_by(sex) %>% slice(1:size)
}


ranking_equality <- function(df, group) {
  classi <- unique(df[[group]])
  len <- length(classi)
  dfs <- c(1:len)*0
```

```

  i <- 1
  for(classe in classi){
    d <- df[df[[group]] == classe ,]
    d <- d[order(-d[[outcome]]) ,]
    dfs[i] <- d
    i <- i+1
  }

  top500 <- head(dfs[1], 500/len)

  top500 <- rbind(head())

  top500 <- head(ord_hr,500)
  top250 <- head(ord_hr,250)
  top100<- head(ord_hr,100)

  newList <- list("top100" = top100, "top250" = top250, "top500"=
    top500)

  return (newList)
}

opportunity_lossrate <- function(df) {

  hr <- df
  tipi = unique(hr$node_placement)

  hr$OpportunityLossRate <- 0

  ### Opportunity Loss Rate
  for (k in tipi){
    om <- mean(hr[which(hr$node_placement == k) ,][[outcome]])
    oms <- mean(hr[which(hr$node_placement == k) ,][["outcome"]])

    index <- hr$node_placement == k
    hr$OpportunityLossRate[index] <- round(oms - om,3)

  }

  ### normalizziamo la variazione di outcome calcolata
  hr$OpportunityLossRate <- normalize_var(hr$OpportunityLossRate
    ,-1,1)
  return(hr)
}

plot_cdf <- function(df, info) {

  hr <- df
```

```
  par(mfrow=c(2,4))

  for(k in 1:nrow(info)){
    tipo <- info[k,]$tipi
    ss <- hr[ hr$node_placement == tipo ,]

    grado <- info[k,]$grado
    yy <- sort(ss[[outcome]])

    ba <- bern_app2(yy, grado)
    plot(ecdf(yy),xlab="outcome", ylab="CDF", main=paste0("CDF
   estimation for type ",tipo))
    lines(yy,ba, col="blue", type="b", lty=2,pch = 18)
    legend("topleft", legend=c("ECDF", "Bernstein approx."),
           col=c("black", "blue"), lty=1:2, cex=0.8)
  }

}

distributive_rate <- function(df) {
  ### Distributive Rate
  df$distributive_rate <- df[[outcome]] - df$outcome
  df$distributive_rate <- normalize_var(df$distributive_rate,-1,1)
  return(df)
}

stampa_dist_rate <- function(rl) {

  for(n in names(rl)){
    rl[[n]]$Ranking <- n
    rl[[n]]$distributive_rate <- rl[[n]][[outcome]] - rl[[n]]$outcome
    rl[[n]]$distributive_rate <- normalize_var(rl[[n]]$distributive_
   rate,-1,1)
  }

  rbl <- rbindlist(rl)

  rbl <- ddply(rbl, .(Ranking), summarize,  Outcome_medio=mean(
   outcome), Distributive_rate_medio=mean(distributive_rate))
  rbl[['Outcome_medio']] <- round(rbl[['Outcome_medio']],3)
  names(rbl) <- c("Ranking","Mean Outcome","Mean Distributive Rate")

  formattable(rbl,
              align =c("l","r", "r"),
              list(`Mean Distributive Rate` = formatter("span",
                                                         x ~ percent(x
   ),
                                                         style = x ~
   style(color = ifelse(x > 0, "green", "red"))),
```

```
                    `Mean Outcome` = color_bar(color = "lightblue", fun=unit.scale)
                    )
               )
}

shannon_diversity <- function(df) {
  n_col <- ncol(df)
  for(col_index in 1:n_col) {
    t <- table(df[,col_index])
    print(diversity(t))
  }
}

stampa_shannon <- function(df) {
  # X

  df$node_placement <- NULL
  df$quantile<- NULL
  df$outcome<- NULL

  b <- df$Ranking
  df$Ranking <- NULL

  Feature <- names(df)

  sh_or <- sapply(df, function(x) diversity(table(x)) )

  s1 <- data.frame(Feature,sh_or)
  s1$Dataset <- b[1:nrow(s1)]

  names(s1) <- c("Feature", "Shannon","Dataset")

  ggplot(s1, aes(fill=Dataset, y=Shannon, x=Feature)) +
    theme(axis.text.x=element_text(angle=90,hjust=1)) +
    geom_bar(position="dodge", stat="identity") +
    labs(y="Shannon Index", x = "Circumstances")
}

stampa_shannon_compare <- function(df) {
  # X

  df$node_placement <- NULL
  df$quantile<- NULL
  df$outcome<- NULL

  Feature <- names(df)

  x1<-split(df, df$sample)
```

```
  sh_or <- sapply(x1$original, function(x) diversity(table(x)) )
  sh_mod <- sapply(x1$modified, function(x) diversity(table(x)) )

  s1 <- data.frame(Feature,sh_or)
  s1$Dataset <- "Setting 1"

  s2 <- data.frame(Feature,sh_mod)
  s2$Dataset <- "Setting 2"

  names(s1) <- c("Feature", "Shannon","Dataset")
  names(s2) <- c("Feature", "Shannon","Dataset")

  s <- bind_rows(s1,s2)
  s <- s[s$Feature != "sample",]
  ggplot(s, aes(fill=Dataset, y=Shannon, x=Feature)) +
    theme(axis.text.x=element_text(angle=90,hjust=1)) +
    geom_bar(position="dodge", stat="identity") +
    labs(y="Shannon Index", x = "Circumstances")
}

stampa_theil <- function(rl) {

  for(n in names(rl)){
    rl[[n]]$Ranking <- n
  }

  rbl <- rbindlist(rl)
  rbl <- ddply(rbl, .(Ranking), summarize,  G3=Theil(G3), Outcome=
    Theil(outcome))

  df2 <- tidyr::pivot_longer(rbl, cols=c('G3', 'Outcome'), names_to='
    variable',
                                          values_to="Theil")

  ggplot(df2, aes( x=Ranking, y=Theil, fill=variable)) +
    geom_bar(position="dodge", stat="identity") +
    labs(y="Theil Index")
}

theil_entropy <- function(df) {
  Theil(df$outcome)
  Theil(df[[outcome]])
}

olp_table <- function(olp) {
  d <- olp$depr$deprivati
  ds <- olp$depr_std$deprivati_std
  f <- olp$fort$fortunati
  fs <- olp$fort_std$fortunati_std
```

```
  Deprivati <- levels(d)[as.numeric(d)]
  Deprivati_standard <- levels(ds)[as.numeric(ds)]
  Privilegiati <- levels(f)[as.numeric(f)]
  Privilegiati_standard <- levels(fs)[as.numeric(fs)]

  df1 <- data.frame(Deprivati, Deprivati_standard, Privilegiati,
    Privilegiati_standard,stringsAsFactors=FALSE )

  return(df1)
}

stampa_tipi <- function (df) {

  tops <- data.frame(xtabs(~Ranking+node_placement, data=df))
  tops <- ddply(tops, .(Ranking), summarize,  Percentage=Freq/sum(
    Freq)*100, Type = node_placement)

  ggplot(tops, aes(fill=Type, y=Percentage, x=Ranking)) +
    geom_bar(position="dodge", stat="identity") +
    scale_y_continuous(breaks = scales::pretty_breaks(n = 10))
}
```

	Min.	X1st.Qu.	Median	Mean	X3rd.Qu.	Max.
age	15	16	17	16.7442219	18	22
Medu	0	2	2	2.5146379	4	4
Fedu	0	1	2	2.3066256	3	4
traveltime	1	1	1	1.5685670	2	4
studytime	1	1	2	1.9306626	2	4
failures	0	0	0	0.2218798	0	3
famrel	1	4	4	3.9306626	5	5
freetime	1	3	3	3.1802773	4	5
goout	1	2	3	3.1848998	4	5
Dalc	1	1	1	1.5023112	2	5
Walc	1	1	2	2.2804314	3	5
health	1	2	4	3.5362096	5	5
absences	0	0	2	3.6594761	6	32
G3	0	10	12	11.9060092	14	19

Figure A.1: Summary statistic of the dataset

Bibliography

[1] *Automated Decision Systems Task Force*. URL: https://www1.nyc.gov/site/adstaskforce/index.page (visited on 12/12/2019) (cit. on p. 1).

[2] Frank Pasquale. *The black box society: the secret algorithms that control money and information*. Cambridge: Harvard University Press, 2015. 311 pp. ISBN: 978-0-674-36827-9 (cit. on pp. 1, 3).

[3] Cathy O'neil. *Weapons of math destruction: How big data increases inequality and threatens democracy*. Broadway Books, 2016 (cit. on pp. 1, 24).

[4] Solon Barocas and Andrew D. Selbst. «Big Data's Disparate Impact». In: *SSRN Electronic Journal* (2016). ISSN: 1556-5068. DOI: 10.2139/ssrn.2477899. URL: https://www.ssrn.com/abstract=2477899 (visited on 03/09/2020) (cit. on p. 1).

[5] Solon Barocas, Moritz Hardt, and Arvind Narayanan. *Fairness and Machine Learning*. http://www.fairmlbook.org. fairmlbook.org, 2019 (cit. on pp. 1, 28).

[6] Jeff Larson Julia Angwin. *Machine Bias*. ProPublica. May 23, 2016. URL: https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing (visited on 11/20/2019) (cit. on pp. 1, 21).

[7] AlgorithmWatch. «Automating Society Report 2019». In: (Jan. 2019), p. 148 (cit. on p. 4).

[8] Thomas Davenport. *Thomas H. Davenport*. type: dataset. Aug. 14, 2019. DOI: 10.1287/8943f842-86f8-4d42-9a64-9a7cd07b31f5. URL: http://pubsonline.informs.org/do/10.1287/8943f842-86f8-4d42-9a64-9a7cd07b31f5/abs/ (visited on 12/12/2019) (cit. on p. 4).

[9] Francesco Ricci, Lior Rokach, and Bracha Shapira. «Introduction to recommender systems handbook». In: *Recommender systems handbook*. Springer, 2011, pp. 1–35 (cit. on p. 4).

[10] P. Bonhard. «Improving recommender systems with social networking». In: *Proceedings Addendum of the 2004 ACM Conference on Computer-Supported Cooperative Work. Chicago, IL, USA*. 2004 (cit. on p. 5).

[11] Sahin Cem Geyik, Stuart Ambler, and Krishnaram Kenthapadi. «Fairness-Aware Ranking in Search & Recommendation Systems with Application to LinkedIn Talent Search». In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '19* (2019), pp. 2221–2231. DOI: 10.1145/3292500.3330691. arXiv: 1905.01989. URL: http://arxiv.org/abs/1905.01989 (visited on 03/17/2020) (cit. on p. 5).

[12] Ashudeep Singh and Thorsten Joachims. «Policy learning for fairness in ranking». In: *Advances in Neural Information Processing Systems*. 2019, pp. 5427–5437 (cit. on pp. 5, 30, 66).

[13] Meike Zehlike, Tom Sühr, Carlos Castillo, and Ivan Kitanovski. «FairSearch: A Tool For Fairness in Ranked Search Results». In: *arXiv preprint arXiv:1905.13134* (2019) (cit. on pp. 5, 66).

[14] Songül Tolan, Marius Miron, Emilia Gómez, and Carlos Castillo. «Why Machine Learning May Lead to Unfairness: Evidence from Risk Assessment for Juvenile Justice in Catalonia». In: *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Law - ICAIL '19*. the Seventeenth International Conference. Montreal, QC, Canada: ACM Press, 2019, pp. 83–92. ISBN: 978-1-4503-6754-7. DOI: 10.1145/3322640.3326705. URL: http://dl.acm.org/citation.cfm?doid=3322640.3326705 (visited on 11/05/2019) (cit. on p. 6).

[15] *Austria's employment agency rolls out discriminatory algorithm, sees no problem*. AlgorithmWatch. URL: https://algorithmwatch.org/en/story/austrias-employment-agency-ams-rolls-out-discriminatory-algorithm/ (visited on 11/05/2019) (cit. on p. 9).

[16] *Black box Schufa*. Data Journalism Awards. URL: https://datajournalismawards.org/projects/black-box-schufa/ (visited on 11/05/2019) (cit. on p. 10).

[17] Dan Sabbagh Defence and security editor security. «Regulator looking at use of facial recognition at King's Cross site». In: *The Guardian* (Aug. 12, 2019). ISSN: 0261-3077. URL: https://www.theguardian.com/uk-news/2019/aug/12/regulator-looking-at-use-of-facial-recognition-at-kings-cross-site (visited on 11/06/2019) (cit. on p. 11).

[18] Adam Vaughan. *UK launched passport photo checker it knew would fail with dark skin*. New Scientist. URL: https://www.newscientist.com/article/2219284-uk-launched-passport-photo-checker-it-knew-would-fail-with-dark-skin/ (visited on 11/05/2019) (cit. on p. 12).

[19] *Prisoner risk algorithm could program in racism*. The Bureau of Investigative Journalism. URL: https://www.thebureauinvestigates.com/stories/2019-11-14/prisoner-risk-algorithm-could-program-in-racism (visited on 11/17/2019) (cit. on p. 12).

[20] Joy Buolamwini and Timnit Gebru. «Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification». In: (), p. 15 (cit. on p. 13).

[21] Maarten Sap, Dallas Card, Saadia Gabriel, Yejin Choi, and Noah A. Smith. «The Risk of Racial Bias in Hate Speech Detection». In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. Florence, Italy: Association for Computational Linguistics, 2019, pp. 1668–1678. DOI: 10.18653/v1/P19-1163. URL: https://www.aclweb.org/anthology/P19-1163 (visited on 11/05/2019) (cit. on p. 14).

[22] «How Amazon's Algorithms Curated a Dystopian Bookstore». In: *Wired* (). ISSN: 1059-1028. URL: https://www.wired.com/story/amazon-and-the-spread-of-health-misinformation/ (visited on 11/05/2019) (cit. on p. 16).

[23] Reuters. «Amazon ditched AI recruiting tool that favored men for technical jobs». In: *The Guardian* (Oct. 10, 2018). ISSN: 0261-3077. URL: https://www.theguardian.com/technology/2018/oct/10/amazon-hiring-ai-gender-bias-recruiting-engine (visited on 11/06/2019) (cit. on p. 17).

[24] Kristian Lum and William Isaac. «To predict and serve?» In: *Significance* 13.5 (2016), pp. 14–19. ISSN: 1740-9713. DOI: 10.1111/j.1740-9713.2016.00960.x. URL: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2016.00960.x (visited on 11/19/2019) (cit. on p. 18).

[25] *Amazon Doesn't Consider the Race of Its Customers. Should It?* Bloomberg.com. URL: http://www.bloomberg.com/graphics/2016-amazon-same-day/ (visited on 11/20/2019) (cit. on p. 20).

[26] Muhammad Ali, Piotr Sapiezynski, Miranda Bogen, Aleksandra Korolova, Alan Mislove, and Aaron Rieke. «Discrimination through optimization: How Facebook's ad delivery can lead to skewed outcomes». In: *Proceedings of the ACM on Human-Computer Interaction* 3 (CSCW Nov. 7, 2019), pp. 1–30. ISSN: 25730142. DOI: 10.1145/3359301. arXiv: 1904.02095. URL: http://arxiv.org/abs/1904.02095 (visited on 11/20/2019) (cit. on p. 23).

[27] Cynthia Dwork, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Richard Zemel. «Fairness through awareness». In: *Proceedings of the 3rd innovations in theoretical computer science conference*. 2012, pp. 214–226 (cit. on p. 25).

[28] Wikipedia. *Garbage in, garbage out — Wikipedia, The Free Encyclopedia.* `http://en.wikipedia.org/w/index.php?title=Garbage%20in%2C%20garbage%20out&oldid=941698693`. [Online; accessed 09-March-2020]. 2020 (cit. on p. 25).

[29] ISO/IEC. *ISO/IEC 25012: Software Engineering - Product Quality.* first ed. Geneva: ISO/IEC, 2008 (cit. on p. 25).

[30] ISO/IEC. *ISO/IEC 25024: Software Engineering - Product Quality.* first ed. Geneva: ISO/IEC, 2015 (cit. on p. 26).

[31] David Camilo Corrales, Juan Carlos Corrales, and Agapito Ledezma. «How to address the data quality issues in regression models: a guided process for data cleaning». In: *Symmetry* 10.4 (2018), p. 99 (cit. on p. 26).

[32] Antonio Vetrò, Lorenzo Canova, Marco Torchiano, Camilo Orozco Minotas, Raimondo Iemma, and Federico Morando. «Open data quality measurement framework: Definition and application to Open Government Data». In: *Government Information Quarterly* 33.2 (2016), pp. 325–337 (cit. on p. 26).

[33] Robyn M. Dawes, David Faust, and Paul E. Meehl. «Clinical versus actuarial judgment.» In: *Science* 243 4899 (1989), pp. 1668–74 (cit. on p. 28).

[34] Reuben Binns. «Fairness in Machine Learning: Lessons from Political Philosophy». In: (2017), p. 11 (cit. on p. 28).

[35] Carlos Castillo. «Fairness and transparency in ranking». In: *ACM SIGIR Forum.* Vol. 52. 2. ACM New York, NY, USA. 2019, pp. 64–71 (cit. on p. 30).

[36] Ke Yang and Julia Stoyanovich. «Measuring fairness in ranked outputs». In: *Proceedings of the 29th International Conference on Scientific and Statistical Database Management.* 2017, pp. 1–6 (cit. on pp. 30, 31).

[37] Meike Zehlike, Francesco Bonchi, Carlos Castillo, Sara Hajian, Mohamed Megahed, and Ricardo Baeza-Yates. «FA*IR». In: *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management - CIKM '17* (2017). DOI: `10.1145/3132847.3132938`. URL: `http://dx.doi.org/10.1145/3132847.3132938` (cit. on p. 31).

[38] L Elisa Celis, Damian Straszak, and Nisheeth K Vishnoi. «Ranking with fairness constraints». In: *arXiv preprint arXiv:1704.06840* (2017) (cit. on p. 31).

[39] Ashudeep Singh and Thorsten Joachims. «Equality of opportunity in rankings». In: *Workshop on Prioritizing Online Content (WPOC) at NIPS.* 2017 (cit. on p. 31).

[40] John E Roemer and Alain Trannoy. «Equality of opportunity». In: *Handbook of income distribution.* Vol. 2. Elsevier, 2015, pp. 217–300 (cit. on p. 31).

[41] Paolo Brunori and Guido Neidhöfer. «The Evolution of Inequality of Opportunity in Germany: A Machine Learning Approach». In: *SSRN Electronic Journal* (2020). ISSN: 1556-5068. DOI: 10.2139/ssrn.3520652. URL: https://www.ssrn.com/abstract=3520652 (visited on 03/02/2020) (cit. on p. 31).

[42] John E. Roemer. «A Pragmatic Theory of Responsibility for the Egalitarian Planner». In: *Philosophy and Public Affairs* 22.2 (1993), pp. 146–166 (cit. on p. 33).

[43] Francisco H. G. Ferreira and Jérémie Gignoux. «The measurement of inequality of opportunity: theory and an application to Latin America». In: *Review of Income and Wealth* 57.4 (2011), pp. 622–657. DOI: 10.1111/j.1475-4991.2011.00467.x. URL: http://dx.doi.org/10.1111/j.1475-4991.2011.00467.x (cit. on p. 34).

[44] Marc Fleurbaey and Vito Peragine. «Ex Ante Versus Ex Post Equality of Opportunity». In: *Economica* 80.317 (2013), pp. 118–130. DOI: 10.1111/j.1468-0335.2012.00941.x. URL: https://doi.org/10.1111/j.1468-0335.2012.00941.x (cit. on p. 34).

[45] Patrizia Luongo. «Chapter 2 The Implication of Partial Observability of Circumstances on the Measurement of IOp». In: *Gabriel Rodríguez, J. (Ed.) Inequality of Opportunity: Theory and Measurement (Research on Economic Inequality)* 19 (2011), pp. 23–49 (cit. on p. 34).

[46] Xavier Ramos and Dirk Van de Gaer. «Empirical approaches to inequality of opportunity: principles, measures, and evidence». In: *ZA Discussion Paper No. 6672* (2012). Available at SSRN: https://ssrn.com/abstract=2096802 (cit. on p. 34).

[47] John E. Roemer and Alain Trannoy. «Equality of Opportunity». In: *Handbook of Income Distribution* 2.2 (2015), pp. 217–300 (cit. on pp. 34, 35).

[48] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. «A Survey on Bias and Fairness in Machine Learning». In: *https://arxiv.org/abs/1908.09635* () (cit. on p. 35).

[49] Michael Veale and Reuben Binns. «Fairer machine learning in the real world: Mitigating discrimination without collecting sensitive data». In: *Big Data & Society* 4.2 (2017). DOI: 10.1177/2053951717743530. URL: https://doi.org/10.1177/2053951717743530 (cit. on p. 35).

[50] Torsten Hothorn, Kurt Hornik, and Achim Zeileis. «Unbiased Recursive Partitioning: A Conditional Inference Framework». In: *Journal of Computational and Graphical Statistics* 15.3 (2006), pp. 651–674 (cit. on p. 36).

[51] Daniele Checchi and Vito Peragine. «Inequality of opportunity in Italy». In: *Journal of Economic Inequality* 8.4 (2010), pp. 429–450. DOI: 10.1007/s10888-009-9118-3 (cit. on pp. 36, 38).

[52] Paolo Li Donni, Juan Gabriel Rodríguez, and Pedro Rosa Dias. «Empirical definition of social types in the analysis of inequality of opportunity: a latent classes approach». In: *Social Choice and Welfare* 44.3 (2015), pp. 673–701. DOI: 10.1007/s00355-014-0851-6. URL: https://doi.org/10.1007/s00355-014-0851-6 (cit. on p. 36).

[53] Paolo Brunori and Guido Neidhöfer. «The Evolution of Inequality of Opportunity in Germany: A Machine Learning Approach». In: *SERIES Working Papers, N.01/2020* 1 (2020). DOI: 10.2139/ssrn.3520652. URL: https://ssrn.com/abstract=3520652 (cit. on pp. 36, 38, 39).

[54] Christian Strasser Helmut ans Weber. «On the asymptotic theory of permutation statistics». In: *Mathematical Methods of Statistics* 2 (1999), pp. 220–250. URL: http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.41.7071 (cit. on p. 36).

[55] Carlo E. Bonferroni. *Teoria statistica delle classi e calcolo delle probabilità*. Florence, Italy: Pubblicazioni del R. Istituto superiore di scienze economiche e commerciali di Firenze, 1936 (cit. on p. 37).

[56] Alexandre Leblanc. «On estimating distribution functions using Bernstein polynomials». In: *Annals of the Institute of Statistical Mathematics* 64 (2012), pp. 919–943. DOI: 10.1007/s10463-011-0339-4 (cit. on p. 39).

[57] Guan Zhong. «Efficient and robust density estimation using Bernstein type polynomials». In: *Journal of Nonparametric Statistics* 28.2 (2016), pp. 250–271. DOI: 10.1080/10485252.2016.1163349 (cit. on p. 39).

[58] Wikipedia. *Distributive justice — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Distributive%20justice&oldid=943955753. [Online; accessed 13-March-2020]. 2020 (cit. on p. 40).

[59] Max O. Lorenz. «Methods of Measuring the Concentration of Wealth». In: *Publications of the American Statistical Association* 9.70 (1905), pp. 209–219 (cit. on p. 42).

[60] Corrado Gini. «Methods of Measuring the Concentration of Wealth». In: *The Economic Journal* 31.121 (1921), pp. 124–126 (cit. on p. 42).

[61] Joseph L. Gastwirth. «The Estimation of the Lorenz Curve and Gini Index». In: *The Review of Economics and Statistics* 54 (1972), pp. 306–316 (cit. on p. 42).

[62] Wikipedia. *Diversity index — Wikipedia, The Free Encyclopedia*. `http://en.wikipedia.org/w/index.php?title=Diversity%20index&oldid=936670647`. [Online; accessed 16-March-2020]. 2020 (cit. on p. 43).

[63] Pedro Conceição and Pedro Ferreira. «The young person's guide to the Theil index: Suggesting intuitive interpretations and exploring analytical applications». In: (2000) (cit. on p. 43).

[64] *Entropy, Redundancy and Inequality Measures*. URL: `http://www.poorcity.richcity.org/` (visited on 03/16/2020) (cit. on p. 43).

[65] Paulo Cortez and Alice Silva. «USING DATA MINING TO PREDICT SECONDARY SCHOOL STUDENT PERFORMANCE». In: (), p. 8 (cit. on p. 47).

[66] *Numerus clausus*. In: *Wikipedia*. Page Version ID: 942702845. Feb. 26, 2020. URL: `https://en.wikipedia.org/w/index.php?title=Numerus_clausus&oldid=942702845` (visited on 03/04/2020) (cit. on p. 47).