ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

SCHOOL OF ENGINEERING AND ARCHITECTURE

DEPARTMENT OF

COMPUTER SCIENCE – SCIENCE AND ENGINEERING – DISI

MASTER'S DEGREE COURSE IN COMPUTER ENGINEERING

DEGREE THESIS

in

INTELLIGENT SYSTEMS

Fairness in Human Resource — Bias Analysis and Mitigation

SPEAKER

Prof. Andrea Borghesi

CORRELATOR

Prof. Roberta Calegari

CANDIDATE

Giulia Vivarelli

IV 03/2025
Academic Year 2023/2024

Summary

1. Introduction

In recent years, artificial intelligence has become an integral part of everyone's life.
days, offering the ability to automate and speed up otherwise complex processes.
Its development has, however, raised concerns about potential discrimination.
towards protected categories, especially in sensitive contexts such as healthcare,
recruiting or insurance.

Although prejudices may seem purely human, it is known that most
machine learning models reflect and even amplify the *biases* of the contexts in which they
are developed.

For this reason the concept of *fairness* has acquired a central role also in the field
technological, in which it is essential to ensure that algorithms do not perpetuate the
inequalities already deeply rooted in society.

There are two main sources of *unfairness: biased* datasets and biased algorithms , which are the result of uninformed design choices. The latter can generate a ripple effect: algorithms
biased generate new *biased* data that negatively influence human decisions and that
they can in turn be used to train other models.

To counteract the effects of *biased datasets,* the most logical strategy would be to use
"neutral" datasets, which do not contain inequalities. However, datasets of size
suitable for training AI models are normally the result of years of
social and cultural evolution, with all the nuances that derive from it. In addition, the majority
part of the protected categories tends to be underrepresented, making it even more
finding balanced datasets is complex.

For this purpose, researchers and companies from all over the world have developed and continue to develop metrics for identifying these anomalies and techniques to mitigate them
effects.

This thesis addresses the issue of *fairness* in the context of recruiting, using real data,
provided by the **Akkodis company.** The dataset is used to train machine models
learning to recognize suitable candidates for a given job position, based on data
provided, such as educational qualification, sector, years of experience.

The aim is to analyse whether the use of AI in this field, in addition to speeding up the *recruiting* process by automating its early stages, may introduce discrimination in comparisons of protected categories. Together with the analysis of the performance of each model, evaluated, through selected metrics, the level of preliminary fairness.

The results obtained are analyzed taking into account the context in which they were collected.

Akkodis is in fact a company that is actively committed to inclusiveness, and this is reflected on its data, which highlights how hiring choices are justified by skills of each candidate.

This project includes the preprocessing of the dataset, imported from an Excel file, training of different types of models, evaluating the level of fairness and the application of different mitigation techniques.

2. AI Fairness

2.1. The concept of Fairness

As often happens with great technological innovations, AI has become an integral part of society and has acquired an important role also in critical sectors, such as healthcare, work and justice. Its diffusion in these areas was driven by the promise of make processes faster, more efficient and possibly more objective.

Although prejudice is often associated with human nature, in contrast, in the common imagination, to the cold neutrality of an algorithm, the facts expose a reality very different. *Machine learning* algorithms are in fact trained to emulate the human behavior and are therefore also subject to *bias.*

For this reason, fairness has also gained importance in the technological field, where a rigorous and universal definition would be necessary. In the AI environment, this is done reference to fairness as **AI Fairness.**

However, this is an abstract and subjective concept, strictly linked to the context, to the values social aspects of users and the specific objectives of a system. In particular, in algorithms In *machine learning, fairness* often conflicts with the required performance, making a compromise between accuracy and impartiality is necessary. Those who develop these systems finds itself having to balance *stakeholder* demands with the requirements imposed by new read about it.

However, impartiality is often put on the back burner in order to give priority to *performance.* Often the target of these applications is not the entire community but a a small subgroup of users, which in most cases does not fall within the protected categories and is more represented in the datasets used for the training. 1

In the field of artificial intelligence , impartial models are defined as *fair ,* which do not produce unfavorable results based on sensitive attributes, such as gender, age, ethnicity. The impact of an *unfair* model can be significant, especially in critical contexts.

[1] Madaio et al., «Assessing the Fairness of AI Systems».

A concrete example is given by Google's job recommendation algorithm , which tended to recommend more lucrative job positions to male candidates. A a similar case then emerged with the selection algorithm developed by *Amazon,* withdrawn due to of its tendency to associate lower scores with female candidates. In this case the disparity has been attributed to the underrepresentation of women within the training dataset.[2]

Discrimination in AI systems can be classified into:

- **Discrimination explainable,** through transparency on the use of attributes, not sensitive, which justify it. For example, analyzing the annual income, it could emerge that women have lower earnings than men. However, This could be justified by the number of working hours, which can change between candidates of different sexes and are sometimes lower for women. In In the case of shorter working hours for women, discrimination is said to be legal and seeking balancing it would lead to reverse discrimination, associating men with a salary lower for the same number of hours.[3]

- **Illegal discrimination,** which is not explainable or justified by non-discriminatory attributes. sensitive. It is further subdivided into:

  - **Direct discrimination:** if it directly uses sensitive information to generate unfavorable results towards protected categories

  - **Indirect discrimination:** if it does not directly use sensitive attributes but the results are their implicit effect. A recruiting algorithm could not have access to a candidate's gender but could infer it through other available information, such as participation in a women-only program.[4]

[2] Mujtaba and Mahapatra, «Fairness in AI-Driven Recruitment».
[3] Holzinger et al., *Towards Explainability for AI Fairness.*
[4] Mehrabi et al., «A Survey on Bias and Fairness in Machine Learning».

To avoid these forms of discrimination it is essential for those who develop these systems consider *fairness* from the very early stages of implementation, with a proactive approach multidisciplinary that also takes into account ethical, legal and social aspects.

2.2. Unfairness Sources

Human biases can be transferred to AI models in several ways, including:

- **Biased Dataset:** If the datasets used during training contain, as often happens, *bias,* these will be incorporated and propagated by the models. A dataset it is considered *biased* when it presents a strong imbalance between the categories, with underrepresented groups, or when it has been manually drawn up and can therefore present errors or subjective biases.

- **Label definitions:** The target definition can influence the level of *fairness* of the algorithm. Sometimes a label represents a much more broad and vague and its assignment varies depending on who makes it. The use of selection criteria that present a strong imbalance towards protected categories can distort the model's predictions. For example, a recruiting model might want to identify “good candidates” for a given job position. However, There are several factors, partly subjective, that define a good candidate. The use of certain features, such as the permanence of a candidate within the company, could penalize certain categories and perpetuate the problem. The attribute in question does not necessarily reflect the goodness of a candidate, as a shorter duration can have different causes. Women historically tend to have a lower tenure, for reasons that may be related to maternity or discrimination. People with disabilities may have shorter stays if the company does not have adequate facilities.

- **Feature Selection:** The selection of features to be used during training can improve the performance of a model by reducing complexity computational and computation times. However, this can lead to inequalities in case the selected attributes do not have an adequate representation for all categories present. A *recruiting* algorithm could for example use the university of origin and favor candidates belonging to the university

from which the current managers come. In case the university in question is not frequented by candidates belonging to protected categories, this could influence the algorithm in making incorrect correlations, discarding equally qualified candidates.

- **Proxy:** Even if a model did not have access to sensitive attributes, it could however learn to infer them through other related features, called Proxies. A an example is given by the recruiting algorithm developed by *Amazon* which, although not including gender, he had learned to infer it through attributes such as college of origin, which could sometimes be reserved for a single gender.

- **Masking:** The deliberate manipulation of data to conceal the presence of *bias,* for for example through the aggregation of data into generic categories, it is called *Masking.* [5]

2.3. Bias Detection

To ensure fair treatment in *AI-driven* hiring , it is important implement methods to identify *biases* learned by models, so that they can be mitigated.

An AI system can be considered fair if it guarantees the same results for individuals. whose only difference is constituted by sensible attributes.

Over the years, several formal definitions have been developed to post-process the level of fairness of an algorithm. The choice of the most appropriate metric depends on the context specific, in which false positives and false negatives can have different impacts.

Some of the most popular metrics are:

- **Equalized Odds:** The true positive rate (TPR) and false positive rate (FPR) must be equal, between different groups identified by sensitive attributes. This means that the model must have uniform accuracy, both in the case of correct classifications that you are wrong. In the context of recruiting this means that:

[5] Mujtaba and Mahapatra, «Fairness in AI-Driven Recruitment».

- Suitable candidates, belonging to different categories, must have the same probability of being correctly classified;

- Unsuitable candidates, with insufficient skills, must also have the same probability of being incorrectly classified as positive.

(= 1| = 0, =) = (= 1| = 1, =) , ÿ {0,1}

(A sensitive attribute, predicted class, Y real class)

Independent studies have shown that the COMPAS model, used in the system The United States Judiciary, to assess the risk of recidivism of prisoners, does not respect this metric.6

- **Equal Opportunity:** relaxation of Equalized Odds, takes into account only the rate of true positives (TPR) and therefore ensures that samples from different groups have equal probability of being positively classified if they actually belong to the class positive. It can be useful when false positives are less critical than false negatives, such as in the field of recruiting.

(= 1| = 0, = 1) = (= 1| = 1, = 1)

- **Demographic Parity:** ensures that the rate of positive classifications (TP + FP) is always the same between different groups, without focusing on the real class of each sample, thus ignoring the accuracy of the model. This metric does not take into account account of the real distribution of classes within the different groups, which could to be biased.

1 $ 2 = 03 = 1 $ 2 = 13

[6] Patalay, «COMPAS: Unfair Algorithm?»

- **Fairness Through Awareness:** This metric requires that similar individuals, defined through a specific similarity metric, have similar results, regardless from the protected category to which they belong. In this case individuals with capacity and similar skills should have the same probability of *matching* with a job position.

- **Fairness Through Unawareness:** requires that the model does not explicitly use sensitive characteristics in the decision making process. By excluding such attributes the model is not able to discriminate based on them. However, this metric has of the limits since there are related attributes that can lead back to the sensitive ones and is therefore not suitable for complex datasets. It excludes direct discrimination but not the indirect one.

- **Treatment Equality:** the ratio between false negatives and false positives must be constant across different groups. This metric is particularly appropriate in contexts where the extent of a misclassification varies greatly depending on whether it is positive or negative. For example, in the case of a bank rejecting a mortgage request (FN) or approving it (FP) has a very different impact and should be balanced equally between different demographic groups.

- **Fairness Test:** requires that the probability score S predicted by a model (S = S(x)) has the same predictive meaning for any group. For each value of S the aforementioned class must be the same.

$$(= 1| = \quad , \quad =) = (= 1| = \quad , \quad =)$$

This apparently obvious criterion is fundamental in areas such as justice. criminal, in which it is important that the provisions have the same meaning, regardless of cultural differences.

- **Counterfactual Fairness:** This metric requires that the outcome of a sample remain unchanged by changing the value of a sensitive attribute. A suitable candidate should remain eligible if in a hypothetical world he had a gender or ethnicity different. It can be more complex to process as it requires reworking by part of the new sample model.

(ˢ !ÿ#( ) = | = , =) = (ˢ !ÿ#$( ) = | = , =)

- **Fairness in Relational Domains:** captures the relational structure of the domain, taking into account not only the attributes but also the social connections and organizational between individuals.

- **Conditional Statistical Parity:** extends *Demographic Parity* by requiring parity between different groups within specific subgroups, identified by non-identifiable attributes sensitive. Ensures that individuals who share relevant characteristics, such as for example the area of study or the level of experience, are treated equally, regardless of sensitive attributes.

These metrics provide rigorous strategies to obtain fairer algorithms but it is
It is important to specify that only in rare cases can they be satisfied simultaneously.
For this reason it is essential to identify the most appropriate metrics, based on
ethical and legal considerations, which are as closely aligned as possible with the values and priorities of the users involved.

2.4. Bias Mitigation

Mitigating disparities in AI systems is a complex and multifaceted challenge. They are
Various approaches have been proposed over the years to address the problem and are currently
There are several toolkits available that developers can use to review, report, and

mitigate discrimination in models. Among the latter are the open source projects *AIF3607* and *Fairlearn8.*

These tools accompany developers throughout the life cycle of models, including metrics for datasets and models, *explainers* and mitigation algorithms.

The main strategies are divided into three categories, depending on the stage of the pipeline: to which the following are applied:

- **Pre-processing:** whether it is possible to modify the training data, removing *bias* or balancing underrepresented classes.
- **In-processing:** whether it is possible to modify the training procedure.
- **Post-processing:** if the algorithm can only be treated as a black box, without being able to modify data or training.9

2.4.1. Pre-processing

A common approach is to reprocess the training data to ensure that it represents the entire population, including historically marginalized groups.

This may involve reweighting , *oversampling, undersampling* and *data augmentation,* through the generation of synthetic data. At this stage, they can also be excluding non-representative attributes for all categories in the dataset. This strategy involves identifying and correcting biases in the data before the model is trained on them.

While this technique has been shown to be effective in mitigating biases in datasets, it does not removes the inherent biases of a model and adjustments may therefore be necessary additional during training.

[7] Aasheim and Hufthammer, «Bias Mitigation with AIF360: A Comparative Study».
[8] Bird et al., «Fairlearn: A Toolkit for Assessing and Improving Fairness in AI».
[9] Bellamy et al., «AI Fairness 360».

2.4.2. In-process

Unlike the previous technique, in-processing methods involve the selection of more suitable models and their optimization during the training phase, by imposing specific fairness requirements .

Optimization can be achieved by applying constraints or by making choices of design specifications.

An algorithm could for example be trained to maximize *accuracy parity,* ensuring that the rate of qualified candidates hired is the same across demographic groups different.

Another approach, recently proposed, is *adversarial debiasing,* in which the network main *(predictor)* maximizes *the accuracy* of its predictions and at the same time minimizes the adversary 's *accuracy .* The adversary network aims to prediction of sensitive attributes based on the predictions of the *main predictor.*

With these techniques the model is forced to take into account underrepresented groups for achieve better performance.

However, these *debiasing* techniques have only recently emerged and require more research to demonstrate their feasibility and efficiency.

Furthermore, it is not always possible to alter the structure of a model, which could in some cases be a pre-existing system already trained. In this case the only applicable technique is the *black box* approach in *post-processing.*

2.4.3. Post-processing

Post-processing techniques focus on modifying the output of already developed models. trained, to meet specific fairness requirements.

This could for example include correcting classification thresholds, sometimes by defining separate thresholds, based on the previously defined metrics mentioned.

The advantage of this approach is the possibility of operating *black boxes,* without modifying the underlying architecture. In addition, explanations of the predictions, which highlight the most influential attributes and promote transparency.

In the context of recruiting, providing an explanation of the results to candidates subjected to a AI selection is crucial to justify unfavorable results and enable future improvement. It has emerged that providing clear feedback can improve significantly the perception of justice by the candidates, during the process of *recruiting.* [10]

However, this approach also has limitations, including reduced performance. Furthermore, it alone cannot mitigate *biases* arising from historical data or unbalanced *datasets .*

2.5.Limits

Despite advances in *bias identification and mitigation techniques,* the problem of AI *fairness* still has several limitations.

First, one of the main obstacles is given by the choice of one or more metrics, which can vary significantly between different applications. The selection of a metric over another may require a trade-off between *performance* and fairness. Furthermore, as of the number of metrics imposed, the level of *fairness* could increase but the complexity computational and model performance may suffer.

Additionally, in the *recruiting field,* each job position is unique, with specific requirements. specific, which may include sensitive attributes. A religious assignment could, for example, requiring details such as religion or gender.11 This makes it difficult

[10] Gilliland, «The Perceived Fairness of Selection Systems».
[11] Mujtaba and Mahapatra, «Fairness in AI-Driven Recruitment».

exclude a priori the use of sensitive attributes for training models
recruiting.

Even at a regulatory level, although the European Union is developing regulations
specific with *the AI Act,* collaboration between international jurisdictions is needed
different, as AI systems often operate globally.
Furthermore, some regulations already in force, such as the GDPR in Europe, limit the use of data
sensitive, essential for monitoring the level of fairness.

Finally, some regulations, such as the *AI Act*[12] itself and the *AI Bias Audit Law of New York*[13], contain ambiguities that allow companies to avoid carrying out rigorous controls
if they declare their applications as low risk. These regulations distinguish
in fact, applications in high-risk applications, which operate in critical sectors and
make critical decisions, and low-risk applications, where the model is
used exclusively as a decision aid.

This raises concerns about the actual effectiveness of the regulations, as the simple
human involvement is sufficient to bypass the required *audits ,* but does not in itself exclude
the presence of discrimination.

[12] «Why the AI Act Fails to Protect Civic Space and the Rule of Law».
[13] «New York Lawmakers Aim to Close Loopholes in NYC's AI Bias Audit Law and Add Teeth to Workplace Protections».

3. Akkodis Dataset

Akkodis is a global consulting firm offering *recruitment* and
training. Since 2022 it has been part of the Adecco group, a world leader in recruitment
of staff. This company stands out in the industry for its commitment to inclusivity.

> *"At Akkodis we celebrate diverse backgrounds and empower unique voices. Our inclusive workplace fosters sense of belonging, broad perspectives, and innovation. With our conscious inclusion training and mentorship programs we foster an equitable workforce where everyone can thrive. Join us and make a difference."*[14]
>
> *Veronique Rodoni*
>
> *Group SVP HR Akkodis*

Its inclusive approach is also reflected in its data, which may be less
influenced by human biases, compared to traditionally used datasets. The results
The results show that using a more equitable dataset upstream reduces the need for
mitigation during training and in *post-processing.*

3.1. Dataset Structure

The company's dataset, previously anonymized, features multiple rows for each
candidate stored.

For each candidate, each line identifies a different step in the recruiting process.
The columns can be divided into three macro categories: **candidate attributes,**
Recruiting process attributes and job **position attributes ,** associated with the
candidate.

[14] «Akkodis.com».

CANDIDATE ATTRIBUTES:

- **ID:** unique identifier
- **Candidate State:** candidate status
  - o *Imported:* candidates imported from external databases, such as Alma Degree. Candidates who maintain this status may never have replied to Akkodis. Some have the event *(Event_Type__Val) CV Request,* which indicates that the recruiter has not yet received the resume.
  - o *First contact:* first contacts with the candidate, normally by telephone. Candidates who maintain this status may have cut off contact with Akkodis or they may not have an adequate resume *(Event_Feedback = Inadequate* CV).
  - o *In selection:* candidates in the selection phase, subjected to the first interviews, for a job position selected from those managed by the company
  - o *QM:* candidates subjected to Qualification Meeting
  - o *Economic Proposal:* candidates who have received an economic proposal by the company
  - o *Vivier:* candidates whose skills were not aligned with the requirements of the position for which they were evaluated, but are considered valid by Akkodis for future opportunities.
  - o *Hired* the candidate was hired by the company that hired Akkodis
- **Age Range:** column categorical containing range Of age [< 20], [20 – 25], [26 – 30], [31 – 35], [36 – 40], [40 – 45], [> 45]
- **Residence:** current residence of the candidate
- **Sex:** gender of the candidate, it admits two values *(Male* | *Female)* and the default value is *Bad.*

- **Protected Category:** indicates whether the candidate belongs to protected categories, specifying the reference article *(articles 1* and 18).
- **TAG:** keywords used by the recruiter.
- **Study Area:** area of study, academic discipline of the candidate.
- **Study Title:** degree or academic qualification obtained

  or *Middle school diploma*

  or *Professional qualification*

  or *High school graduation*

  or *Three-year degree*

  or *Five-year degree*

  or *Master's degree*

  or *Doctorate*

- **Years Experience:** range of years of experience of the candidate

  [0], [0-1], [1-3], [3-5], [5-7], [7-10], [+10]
- **Sector:** sector in which the candidate has experience.
- **Last Role:** the candidate's last work or study role.
- **Year of Insertion:** year in which the candidate was inserted into the database.
- **Year of Recruitment:** year in which the candidate was hired, present only if *Candidate State = Hired.*
- **Current Ral:** current RAL of the candidate.
- **Expected Ral:** the candidate's expectation of future RAL.

PROCESS ATTRIBUTES:

They are related to a specific stage and change for the same candidate as he progresses. forward in the recruiting process .

- **Event_Type__Val:** Specifies the type of event, the stage of the recruitment process. The events present in the database can be divided into 3 macro categories:
  - **Initial events:** *Commercial note, CV Request, Contact note, Research Association*
  - **Central events:** *HR interview, BM interview, Technical interview, Qualification Meeting*
  - **Final events:** *Candidate notification, Sending SC to customer, Economic proposal, Notify candidate, Inadequate CV*
- **Event_Feedback:** feedback associated with a specific event *(Event_Type__Val),* can be OK or KO, with any comments specified in brackets. Not all types of events provide feedback.
- **Overall:** score associated with the interview, present only for rows containing *Event_Type__Val* central, associated with interviews.
- **Akkodis headquarters:** Akkodis headquarters that manages the candidate.

Scores assigned by the recruiter, from 1 to 4, during the interview:

- **Technical Skills:** technical skills.
- **Standing/Position:** position within the organization.
- **Communication:** communication skills.
- **Dynamism:** level of dynamism.
- **Mobility:** mobility.
- **English:** English level.

JOB POSITION ATTRIBUTES:

These fields are present only if the candidate has been hired by the company in question.

- **Recruitment Request:** company request for a candidate.
- **Assumption Headquarters:** location of the job position.
- **Job Family Hiring:** job position category.
- **Job Title Hiring:** specific title of the position.
- **Job Description:** description of the role.
- **Candidate Profile:** ideal candidate profile, required by the company.
- **Years Experience.1:** years of experience required, expressed in ranges compatible with the candidate's *Years Experience* field .
- **Minimum Ral:** minimum expected RAL.
- **Ral Maximum:** maximum expected RAL.
- **Study Level:** level of study required for the job position, the values are compatible with the *Study Title field.*
- **Study Area.1:** specifies the required area of study, contains compatible with *Study Area.*
- **Linked_search_key:** field containing a non-unique code in the format *RSnn.nnnn,* where the first two-digit number identifies the year of insertion of the job position while the number after the dot indicates the number of searches made for a specific position.

3.2. Preprocessing

3.2.1. Overview

The dataset features approximately 12,300 candidates, of which only 4% have *Candidate State Hired* and then presents the information regarding the associated job position. After a further analysis revealed that more than 50% of the candidates are *entry* level imported from external databases, for which several fields are missing *(Candidate State = Imported).* In fact, most of these candidates have not had direct contact with Akkodis. This data distribution required a preprocessing phase preliminary to standardize the data.

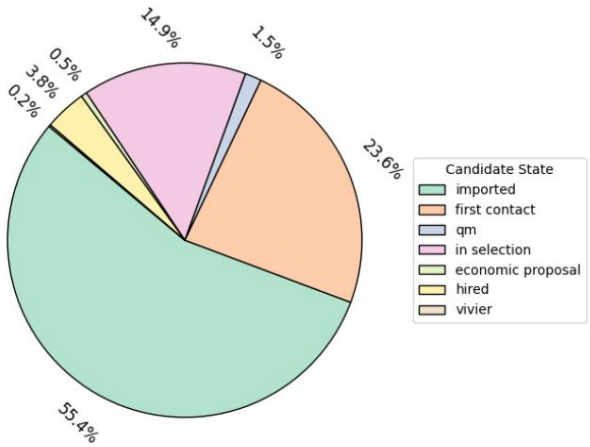

Figure 1 Candidate State Distribution in the Initial Dataset

3.2.2. Data Cleaning

During this first phase, the distribution of missing values was analyzed, for a better understanding of the initial structure of the dataset. This analysis allowed a clear distinction between the three main categories of attributes present:

- Candidate-related attributes
- Attributes related to the *recruiting* process
- Job-related attributes

Subsequently all missing values were temporarily replaced with the value *"not specified",* to preserve the original structure without introducing distortions. The values numerics have been converted to the correct format, while they have been transformed through statistical operations only in the subsequent phases.

To ensure greater uniformity of the text fields, a *mapping* has been applied for synonyms and lexical variants, also correcting any typing errors.

3.2.3. Feature Mapping

Some fields have been reworked to extract useful information and simplify their use. content.

Protected Category

The *Protected Category* field contained the reference article indication *(article 1* | *Article 18)* for candidates belonging to protected categories. This field has been reworked into a binary variable *(yes* | *no).*

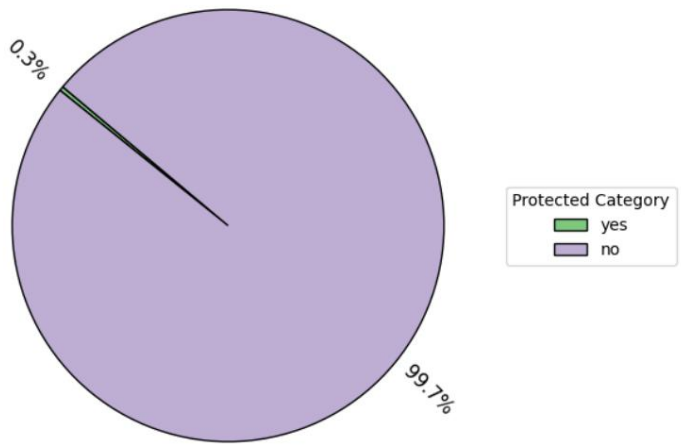

Figure 2 Distribution of Protected Category in the initial dataset

Residence

The *Residence* field contained unstructured data:

```
['turin » turin ~ piedmont' 'conversano » bari ~ puglia'
'caserta » caserta ~ campania' ...
'san felice a cancello » caserta ~ campania'
'perdifumo » salerno ~ campania'
'palmanova » udine ~ friuli venezia giulia']
```

Figure 3 Initial contents of the Residence column

To make the information more organized, four new columns have been extracted:

Residence Country, Residence Italian Region, Residence Italian Province, Residence Italian City.

Id	Residence	Residence Country	Residence Italian Region	Residence Italian Province	Residence Italian City
15	squinzano » lecce ~ puglia	italy	puglia	lecce	squinzano
36	alessandria » alessandria ~ piedmont	italy	piedmont	alessandria	alessandria
39	bari » bari ~ puglia	italy	puglia	bari	bari
41	perch dosimo » cremona ~ lombardy	italy	lombardy	cremona	perch dosimo
47	germany » (state) ~ (overseas)	germany	not in italy	not in italy	not in italy
...	...	...	...	...	...

Figure 4 Detail of columns Residence, Residence Country, Residence Italian Region, Residence Italian Province, Residence Italian City

It is likely that the *Residence* field has been machine translated, for this reason some extracted cities are non-existent. This anomaly is addressed in later stages.
Two additional binary columns have also been introduced: *Italian Residence* and *European Residence.*

Last Role

The Last Role column contained numerous terminological variants for similar positions. For To reduce fragmentation and ensure consistency, a mapping was applied to to make it uniform.

Figure 5 Word Cloud of the top 20 Last Role in the initial dataset

3.2.4. Duplicate IDs

The database provided by **Akkodis** is the result of the fusion of two different databases, belonging to the two companies AKKA and Modis, which were merged by Adecco in 2022. 15

For this reason some IDs are duplicated and belong to different candidates within of the dataset. A useful field for identifying duplicate IDs is given by *Year of Insertion,* which represents the year the candidate was entered into the database and should so be constant.

The process of assigning a new ID was carried out on the basis of the information provided by Akkodis, comparing the values of invariant fields, such as *Year of Insertion, Sex, Age Range etc.*

[15] Adecco Group, «The Adecco Group Completes Majority Acquisition of AKKA Technologies».

The owner company said that none of its employees ever requested a gender change within the database. However, there are more than 300 IDs for which the field *Sex* has more than one meaning. This incongruity can have two possible reasons:

- During the merger different candidates with different genders kept the same ID;
- It is not possible to leave the *Sex* field unspecified if it is unknown by the recruiter and the default expected value is *bad.*

During this phase, more than 1,000 new IDs were generated, bringing the total number of 12,300 to 13,350 distinct candidates.

3.2.5. Dataset Sorting

For each categorical variable an order has been defined, both for visualization reasons and to reorder the dataset. In particular:

- **Candidate State:** the assigned order matches the one provided by Akkodis, according to the internal recruiting system

  'imported' > 'first contact' > 'in selection' > 'qm' > 'vivier' > 'economic proposal' > 'hired'

- **Age Range:**

  '< 20 years' > '20 - 25 years' > '26 - 30 years' > '31 - 35 years' > '36 - 40 years' > '40 - 45 years' > '45 years'

- **Years Experience:**

  'not specified' > '[0]' > '[0-1]' > '[1-3]' > '[3-5]' > '[5-7]' > '[7-10]' > '[+10]'

- **RAL:**

  'not specified' > '-20k' > '20-22k' > '22-24k' > '24-26k' > '26-28k' > '28-30k' > '30-32k' > '32-34k' > '34-36k' > '36-38k' > '38-40k' > '40-42k' > '42-44k' > '44-46k' > '46-48k' > '48-50k' > '+50k'

3.3. Target Selection

The aim of the project is to develop AI models, to be used in the preliminary phases of the recruiting process , to analyze the biases that emerge. The choice of the target was influenced by the structure of the dataset, which contains the candidates and positions managed by Akkodis.

Based on the available data, two possible targets for prediction have been identified automatic:

- **RAL:** a new column for predicting the RAL most suited to the profile of the candidate.

- **Hired:** a new column that labels candidate-position pairs as positive or negative, defining whether the candidate's profile is adequate to the requests of the agency.

The first hypothesis was discarded since more than 90% of the candidates did not present any value for none of the fields associated with the RAL.

On the other hand, in order to distinguish between suitable and unsuitable candidates, it is necessary to analyze the three main types of candidates present:

- **Candidates in the early stages of recruiting:** candidates of whom there are not enough information, as it is still in the preliminary stages.

- **Candidates evaluated for a job position but not hired:** in this case the job position for which they were evaluated is not present in the dataset. This typology is further subdivided into:

    - **Candidates who are suitable** for the job position, who have not been hired for motivations independent of skills.

    - **Candidates who are unsuitable** for the position in question. This type can assume a new *Vivier* value for *Candidate State,* if the candidate is considered potentially valid for future offers and is therefore maintained within the dataset.

- **Hired candidates:** candidates in *Hired status,* have all the relevant information to the job position.

To ensure consistency, it is necessary to remove candidates who would be labeled positively but which do not present the associated job position. Among these are the candidates in the very early stages of recruiting, not yet evaluated for a position, and candidates who for unrelated reasons did not proceed further in the process hiring.

3.3.1. Removing Candidates in the Early Recruiting Stages

Through an analysis of the values of *Event_Type__Val* and *Event_Feedback* for each *Candidate It* is possible to isolate candidates in the early stages of the recruitment process:

- ***Imported:*** This status represents candidates imported from other databases. The type of associated event *(Event_Type__Val)* is normally *CV Request* or *Contact Note.* This typology was entirely discarded for the purposes of the project.

- ***First Contact:*** This status represents candidates contacted by Akkodis, who do not have yet to interview. All candidates have been filtered for this status with only one initial event and which did not feature the *Sector field,* which was chosen as a minimum requirement necessary to be able to evaluate a candidate's profile.

- ***In Selection:*** This status represents candidates in the evaluation phase. The most *BM interviews* and HR interviews are frequent . In this case too, they are filtered candidates who present only initial events.

- ***QM:*** This status represents candidates who have undergone qualification meeting.

- ***Vivier:*** This status represents candidates who have not been hired but have been deemed valid by Akkodis for future opportunities.

- ***Economic Proposal:*** This status represents applicants who have received a economic offer from the company. In this case in the *Event_Feedback* field

there is a reason why the candidate is still in this state and not in the state *Hired.* Some values present are for example *'OK (other candidate)', 'KO (proposed renouncement)', 'KO (ral)' etc.*

- ***Hired:*** This status represents candidates who have successfully completed the recruitment process and have been hired for the job position specified in the dedicated fields.

During this phase, approximately 9,000 candidates were removed.

3.3.2. Removing Valid Candidates

To ensure data consistency it was necessary to remove all samples that would have were eligible but, for independent reasons, were not hired. The criterion of selection was the last feedback received by the candidate and the last associated event.

For this purpose, all the *Event_Feedbacks* that highlight a positive evaluation of the profile or a possible negative outcome independent of the skills:

- ***OK (other candidate):*** the candidate was not hired despite the feedback positive *(OK),* because someone else has been selected. It is not correct to label negative profiles that received this feedback.

- ***KO (lost availability):*** the candidate is no longer available, it is not consistent to assign a negative evaluation of his profile.

- ***OK (hired):*** the candidate has been hired, the database may not have updated the his *Candidate State* if this is different from *Hired.*

- ***OK (waiting for departure):*** the candidate has received a positive response from the company, the assignment is about to begin.

- ***KO (opportunity closed):*** the candidate was not hired because the opportunity job for which he was contacted is no longer available. His skills I am not the cause of his *Candidate State.*

- ***KO (retired):*** the candidate has withdrawn.
- ***KO (ral):*** the candidate has renounced the opportunity for economic reasons.
- ***KO (proposed renunciation):*** the candidate has renounced the opportunity.

All candidates who had received feedback as their last, in the last round, have been removed. associated line, one of the above.

Furthermore, all those who presented a final event were filtered out from the remaining candidates. among the following:

- ***Economic Proposal:*** the candidate has received an economic proposal from the company; therefore, his skills meet the requirements of the position.
- ***Candidate Notification:*** This event represents the last communication with the candidate after the company has made the financial proposal.

During this phase, approximately 500 candidates were filtered.

3.4. Creating the final dataset

The initial dataset provided by Akkodis contains only the job positions of the candidates hired, while there is no data specified for candidates with a different *Candidate State* from *Hired.* For training purposes, a new dataset was generated to host pairs *(candidate, position),* labeled via a new binary column *Hired.*

For this purpose, all job positions were collected in an auxiliary *dataset* , which It was in turn subjected to analysis and *pre-processing* to exclude duplicate positions. Each job position has been assigned a unique ID to simplify the process subsequent.

For each candidate a single row was kept by removing the columns *Event_Type__Val* and *Event_Feedback* and performing a numerical average for the columns of scores obtained during different interviews.

All the candidates hired, with their relative positions, were added to the final dataset associated jobs, positively labeled *(Hired=1).*

The negative pairs were artificially generated through the use of a function of similarity based on *cosine similarity,* applied to the vectors generated with *TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer.* The vectors used were obtained through the conjunction of the text columns:

- *Tag, Study Area, Sector, Last Role* for the candidate.
- *Recruitment Request, Job Family Hiring, Job Title Hiring, Job Description, Candidate Profile, Study Area.1* for the job position.

This approach has allowed us to systematically select more or less similar positions to the sample profile, simulating the real *recruiting process.*

For each candidate hired, an additional line was introduced, labelled negatively, with a job position chosen randomly among the least similar positions to the candidate in question.

For each candidate not hired, two negative lines were added with:

- A job position selected among those most similar to the candidate's profile, for simulate a real candidacy.
- One of the least compatible job positions, to introduce a more open comparison net.

Using the similarity function for position selection simulates the real process of *recruiting,* in which candidates are evaluated for positions compatible with their profile. Using a small number of negative lines for each candidate limits the imbalance of the target column and reflects the nature of the *dataset,* in which most of the candidates were evaluated for a single position among those available.

The final dataset thus generated contains approximately 8,400 rows.

3.4.1. Accessory Columns

Auxiliary columns have been added to the resulting dataset to represent numerically the relationships between the candidate and job position attributes.

Similarity columns, based on cosine similarity:

- *Similarity_Score:* Measures the similarity between the candidate profile and the position working, based on the descriptive columns. This value was used for select positions in the negative pair generation phase.
- *Similarity_Score_Last_Role:* Measures the similarity between the last role played by the candidate *(Last Role)* and the job title *(Job Title Hiring),* assuming that candidates with similar recent experience are at an advantage.
- *Similarity_Score_Study_Area:* Measures the similarity between the study area of the candidate *(Study Area)* and the requested one *(Study Area.1).*

Comparison columns for ordinal attributes:

For each pair of corresponding fields, such as *Study Title* and *Study Level,* it was a scale of ordered levels was defined and the score was calculated according to the formula:

$$= \_\_\_\_\_$$

Where And represent the maximum and minimum values respectively in the level scale associated with attributes.

The scores thus obtained are normalized between -1 and +1:

- A **null value** indicates a perfect match, the candidate meets the requirements required.
- A **positive value** indicates that the candidate has higher requirements than those required.

- A **negative value** indicates that the candidate has insufficient requirements with respect to those requested.

Derived columns include:

- *Study_Title_Score:* Measures the match between the candidate's educational qualifications *(Study Title)* and the level required by the position *(Study Level).*
- *Years_Exp_Score:* Measures the correspondence between the candidate's years of experience *(Years Experience)* and those required *(Years Experience.1).*
- *Ral_Score:* Compares the candidate's salary expectations with the minimum RAL offered by the job position. In this case the formula provides the difference between the level of the RAL offered by the company and that of the candidate. A negative value indicates that the RAL requested by the candidate is higher than the minimum offered by the position, which is therefore insufficient compared to expectations.

Geographic distance column:

- *Distance (km):* Represents the distance between the residence of the sample and the company headquarters *(Assumption Headquarters).* It was obtained using two auxiliary datasets, *italy_geo.xlsx* 16 *and countries.csv* 17, containing the latitude and longitude respectively for Italian cities and foreign countries. In this context all the Italian cities have been translated through an *ad hoc mapping.* After having calculated the corresponding coordinates the distance was calculated with the function *geodesic* from the Geopy library .

[16] HenryChinaski, «Italian-Municipalities-2018-Sql-Json-excel».
[17] «Countries.csv».

4. Training

4.1. Preprocessing

The dataset generated in the previous phases was subjected to further preprocessing, for ensure correct and suitable data for training models in subsequent phases.

Reduction of columns

Redundant, unstructured or unavailable columns have been removed in the early stages of the selection process, to avoid *data leakage* and reduce the dimensionality of the dataset.

Some examples:

- *Candidate State:* Final status of the candidate, not available before the application process selection, when using a prediction model.
- *Job Description:* Too long and structured field, used during processing of the Similarity_Score column .
- *Years Experience.1:* Field represented through the *Years_Exp_Score column,* which relates it to the candidate's *Years Experience .*

```
columns_to_drop = ['Candidate State', 'Tag' , 'Recruitment Request', 'Last Role',
                   'Job Description', 'Candidate Profile', 'Year Of Insertion', 'Year Of Recruitment',
                   'Job Title Hiring', 'Years Experience.1', 'Study Level', 'Study Area.1',
                   'Job ID', 'Residence_coord', 'Assumption_coord', 'Assumption Headquarters', 'Residence Italian City',
                   'Candidate_Info', 'Position_Info', 'Akkodis Headquarters', 'Job Family Hiring', 'Linked_Search__Key']
```

Figure 6 Columns removed in pre-processing

Categorical Attribute Encoding

The categorical columns have been converted into numeric format, through *encoding,* for enable their use by machine learning models.

Ordinal variables, such as *Age Range,* were mapped onto a numerical scale consistent with their meaning.

The nominal categorical variables were instead converted using the LabelEncoder of *Scikit-Learn,* which assigns a unique integer to each category.

Missing values

Missing numeric values have been replaced with the mean of the respective column. The values categorical ones have instead maintained *'not specified'* as a separate category, codified also in the previous phase.

Standardization of numerical features

The following numeric columns have been standardized, using the StandardScaler of *Scikit-Learn,* which uses the mean and standard deviation of each column to transform its values.

```
columns_to_standardize = ['Years Experience', 'Minimum Ral', 'Ral Maximum', 'Current Ral',
                          'Expected Ral', 'Distance (km)']

scaler = StandardScaler()
```

Figure 7 Standardized columns in pre-processing phase

This ensures that all numeric fields are at the same scale and therefore have the same impact on the model.

4.2. Dataset analysis

Before training, a graphical analysis of the main features was performed, with emphasis on sensitive attributes and their relationship to other relevant attributes.

4.2.1. Correlation matrix

To highlight the relationships between the columns of the dataset and obtain a graphical representation summary correlation matrix was generated, using the heatmap function of *Seaborn.*

The image, in Figure 8, highlights some relationships, including:

- The **target** column *Hired* shows a correlation with *Similarity_Score;*

- *Years Experience* highlights expected correlations with *Current Ral, Expected Ral* and *Technical Skills, Years_Exp_Score* and *Age Range_encoded;*

- The scores obtained during the interviews are correlated with each other: *Overall, Technical Skills, Standing/ Position, Communication, Maturity, Dynamism;*

- The minimum RAL foreseen by the company *Minimum Ral* shows a correlation with *Years_Exp_Score,* highlighting how its value is related to the years of experience required by the company;

- The candidate's RAL fields *Current Ral* and *Expected Ral* are also correlated with *Ral Score,* which represents the relationship between the candidate's expectations and the proposal company economics;

- The *Distance (km)* field shows expected correlations with the residence fields: *European Residence_encoded, Italian Residence_encoded.*

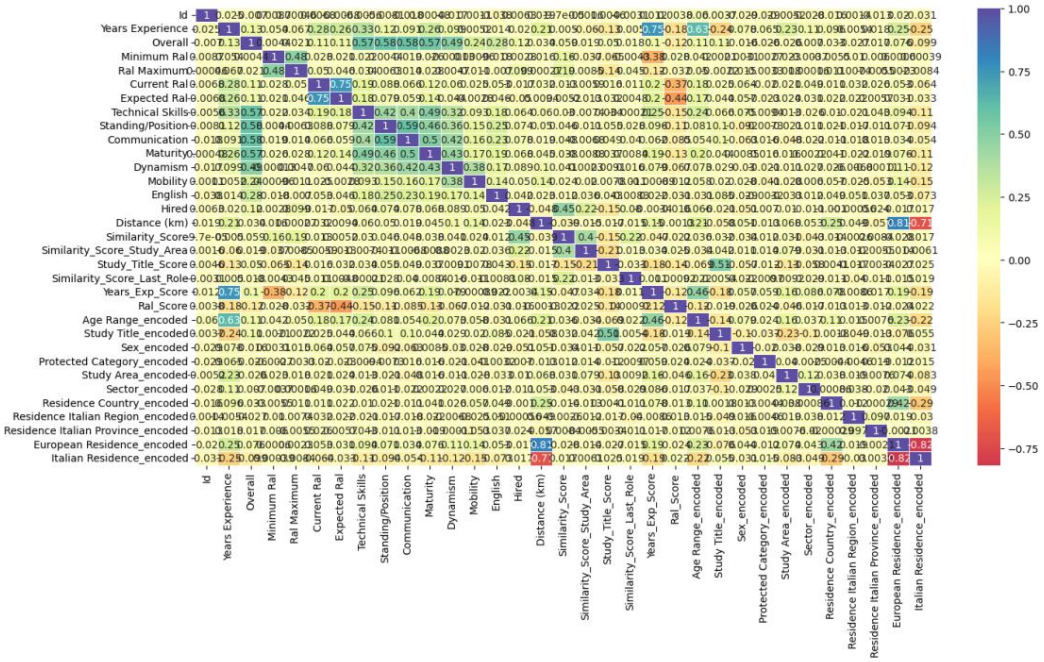

Figure 8 Correlation Matrix

4.2.2. Target Column

The final distribution of the target column, represented by the pie chart in Figure 9, is very unbalanced, highlighting how only 6% of the *entries* are labelled positively, through the *Hired field.*

It will be necessary, during the training phase, to balance the dataset through synthetic samples, to ensure that models do not only predict negative classes.

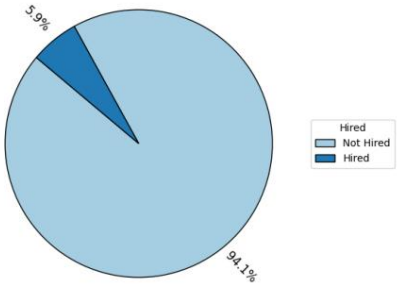

Figure 9 Hired Distribution

4.2.3. Sensitive Attributes Analysis

The distribution of sensitive attributes and their relationship with fields was analyzed. determinants such as the *Hired* **target** column, *Study Title_encoded* educational qualifications , years of experience *Years Experience,* the maximum expected RAL *Ral Maximum* etc.

Sex

The dataset is very unbalanced, with only 20% of candidates belonging to the category *female,* highlighting an underrepresentation.

However, analyzing the distribution of the *Hired* label within each subgroup of *Sex* emerges that, in percentage terms, women have been hired more. In particular, it turns out that 8.3% of women were hired compared to 5.3% for men.

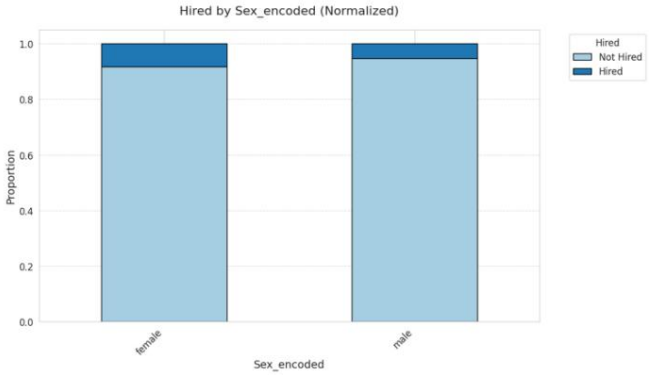

Figure 10 Normalized Hired distribution for each category (female|male)

This slight difference could however find justification in the skills of the candidates, as can be seen from the distribution of educational qualifications in the two categories, in Figure 11. The women in the dataset tend to have higher educational qualifications, such as *five-year degree, master's degree* and *doctorate.*

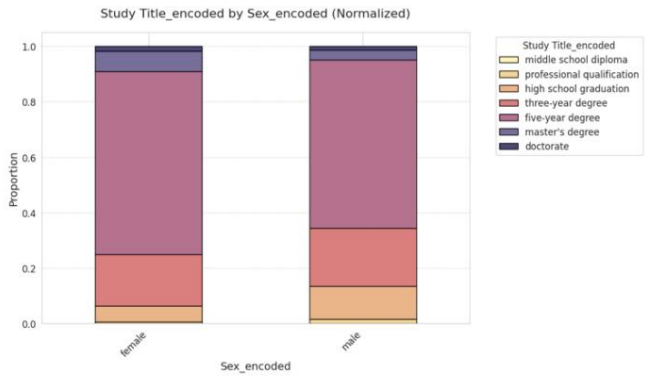

Figure 11 Normalized Study Title_encoded *distribution for each category (female|male)*

However, analyzing the values associated with the *Ral Maximum column,* for the hired candidates, a contrasting difference emerges: with the same degree *(Study* Title) and years of study experience *(Years* Experience) women tend to have higher salaries (Ral *Maximum)* low (Figure 12, Figure 13).

On the other hand, not having sufficient information on the number of hours or the type of contract, It is not possible to justify or condemn these differences.

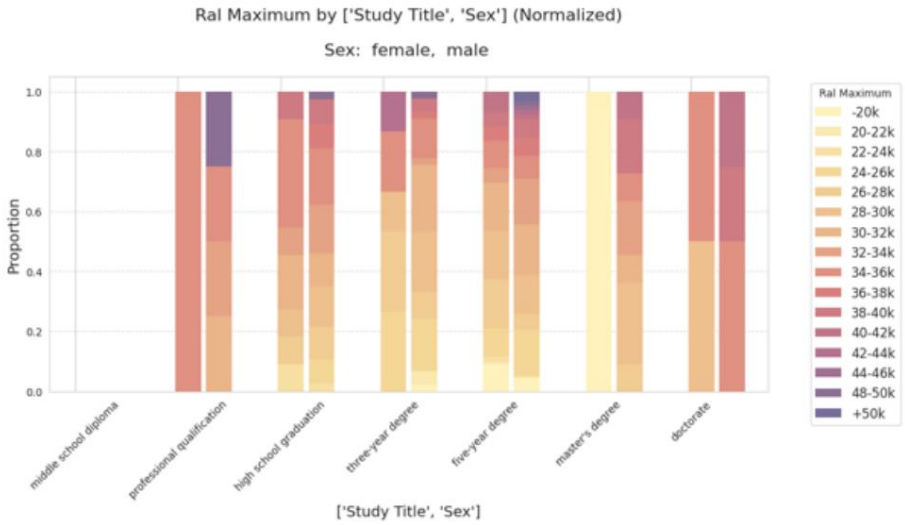

Figure 12 Ral comparison between Sex subgroups with equal Study Title

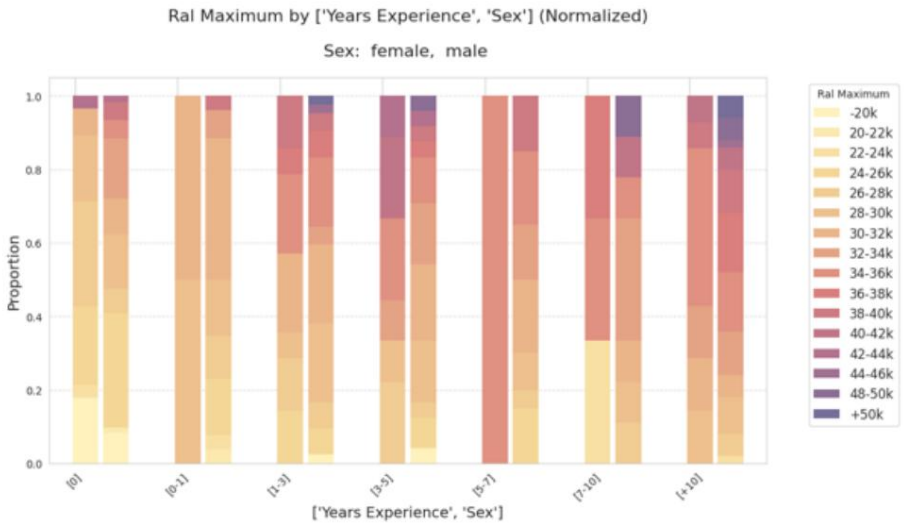

Figure 13 Ral comparison between Sex subgroups with equal Years of Experience

Protected Category

The dataset is heavily skewed towards the *Protected Category field,* with only the 0.6% composed of candidates belonging to protected categories.

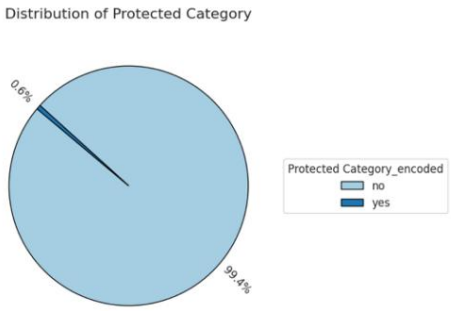

Figure 14 Protected Category Distribution

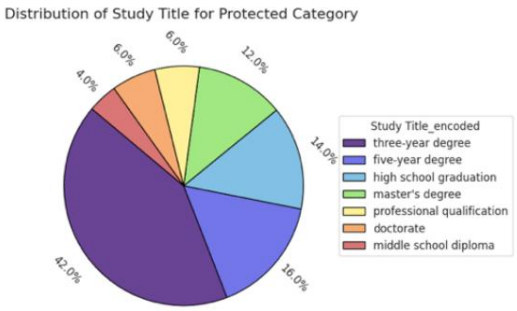

Figure 15 Distribution of qualifications for candidates belonging to protected category

The small number of candidates belonging to a protected category, equal to eighteen, does not give the possibility of making an accurate comparison with equal skills.

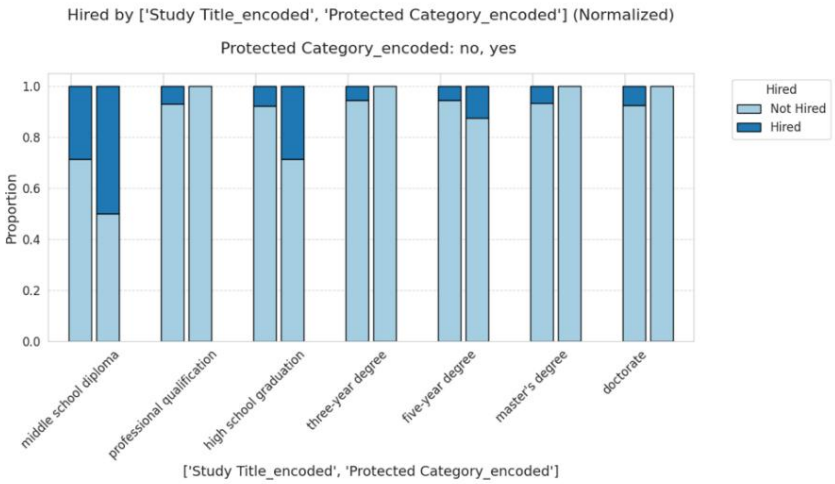

Figure 16 Hired distribution comparison by Protected Category subgroups, with equal Study Title

Age Range

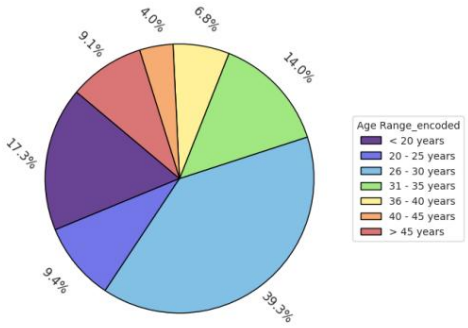

Figure 17 Age Range Distribution

More than 65% of the dataset consists of candidates under the age of 30, with 17% of the age less than 20. However, even if in smaller numbers, the remaining categories have a rate of higher intake.

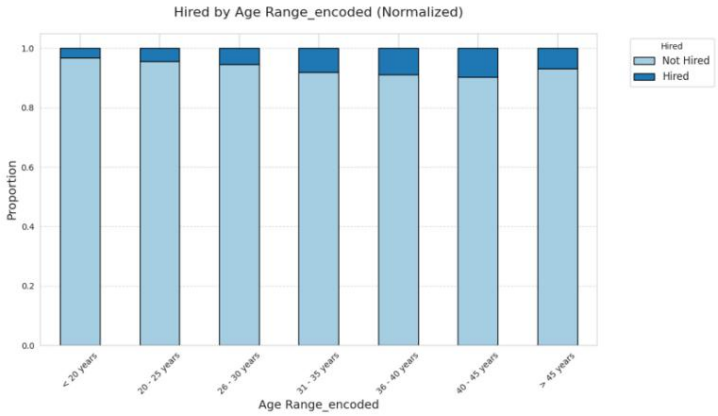

Figure 18 Hired distribution for each Age Range subgroup

Residence

Most of the candidates reside in Italy. Among the candidates with non-Italian residency, the most of which 25% reside in Oman.

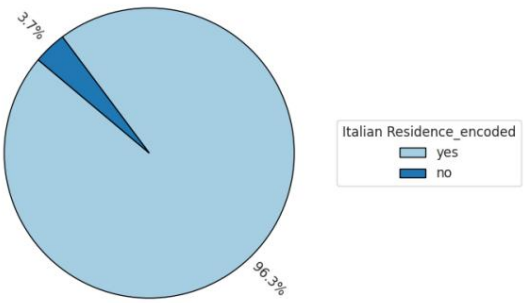

Figure 19 Distribution of Italian Residence

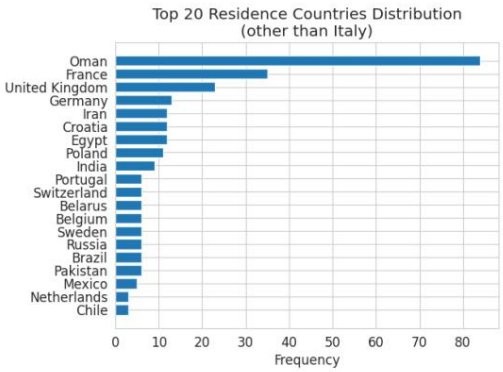

Figure 20 Top 20 Countries of Residence (Italy excluded)

Within the dataset, 6% of the samples residing in Italy are employed, compared to 4% for non-resident candidates.

However, analyzing the other attributes it emerges that candidates who are not residents in Italy have in percentage of higher education qualifications *(Study Title)* and more years of experience *(Years Experience).* However, for the same educational qualification, the distribution of *Ral Maximum* varies significantly between resident and non-resident candidates.

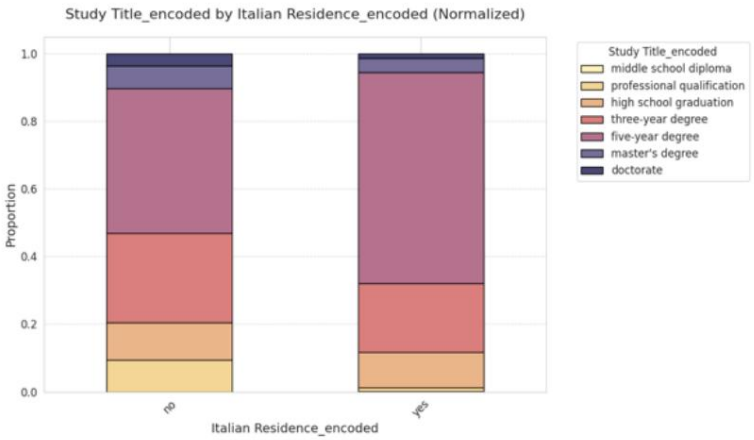

Figure 21 Study Title Distribution for Non-Resident and Resident Candidates in Italy

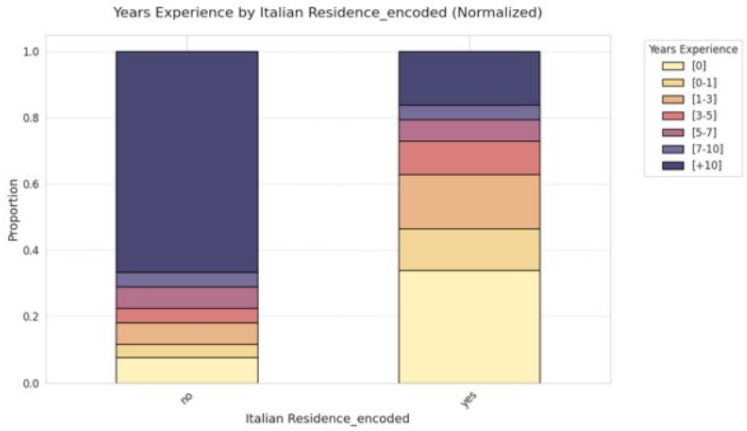

Figure 22 Distribution of Years of Experience for non-resident and resident candidates in Italy

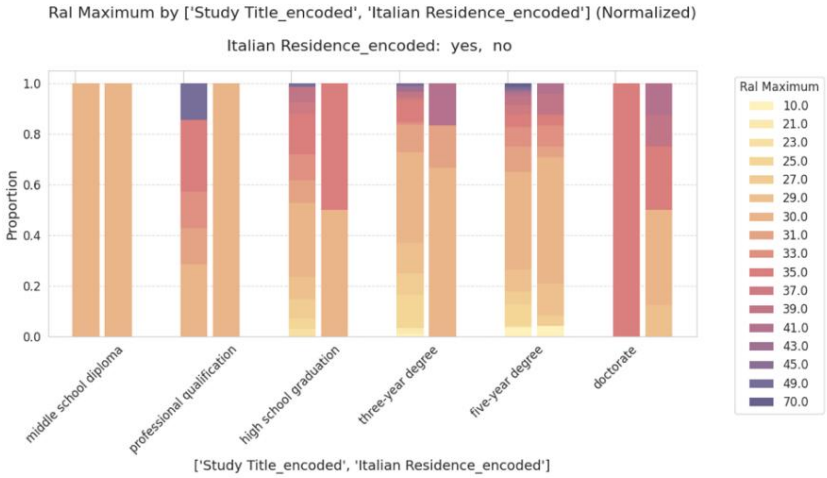

Figure 23 Maximum Ral distribution for candidates resident and non-resident in Italy, with the same educational qualification

4.3.Choice of models

To ensure a comprehensive overview, models belonging to
different algorithmic families, including linear, probabilistic, tree and network models
neural. The goal is to compare results and relate *performance* and *fairness.*

Selected models include:

- *Linear Models*
  - or Linear Regression
  - or Logistic Regression
  - or Linear Perceptron
- *Probabilistic Models*
  - or Gaussian Naïve Bayes
- *Tree-based models*
  - or Decision Tree
  - or Random Forest
  - or AdaBoost
- *Distance-based models*
  - or K-Nearest Neighbors
- *Neural Networks*

4.3.1. Neural Network

In order to be able to use standard tools in the subsequent phases, such as GridSearchCV, the network
neural was realized through an ad hoc class, CustomKerasClassifier, which
extends Scikit-Learn 's BaseEstimator and ClassifierMixin .

The model adopted for neural networks is a *fully connected network,* created using *Tensorflow* and *Keras.*

```
class CustomKerasClassifier(BaseEstimator, ClassifierMixin):
    def __init__(self, input_shape, neurons=10, activation='relu', optimizer='adam',
                 batch_size=32, epochs=20, learning_rate=0.001):
        ...

    def build_tf_model(self):
        model = Sequential([
            Input(shape=self.input_shape),
            Dense(self.neurons, activation=self.activation),
            BatchNormalization(), Dropout(0.4),
            Dense(self.neurons, activation=self.activation),
            BatchNormalization(), Dropout(0.4),
            Dense(self.neurons // 2, activation=self.activation),
            BatchNormalization(),
            Dense(1)
        ])

        optimizer_fn = {'adam': Adam, 'rmsprop': RMSprop, 'sgd': SGD}.get(self.optimizer)
        if optimizer_fn is None:
            raise ValueError("Unknown optimizer")

        model.compile(optimizer=optimizer_fn(learning_rate=self.learning_rate),
                      loss=BinaryCrossentropy(from_logits=True), metrics=['accuracy'])
        return model

    def fit(self, X, y, validation_data=None, early_stopping=None):
        ...
```

Figure 24 CustomKerasClassifier pseudo class code

The architecture, as shown in Figure 24, is composed of:

- An *input layer* whose size matches the number of features in the dataset.
- Three *Dense hidden layers,* where the activation function and the number of neurons are parameters to optimize during *Gridsearch.* The third layer uses half of the neurons *(neurons // 2).*
- Each hidden layer is followed by *Batch Normalization,* to increase stability, e *Dropout* to 40% to reduce overfitting.
- *The output layer* is composed of a single neuron, with no activation function Sigmoid. There function Of loss used And there BinaryCrossentropy(from_logits=True), which incorporates the sigmoid into more stable way.

The optimizer and *learning rate* are also parameters selected in the subsequent stages. through *Gridsearch.* The metric used is *accuracy.*

The model also uses the class_weight parameter in the fit() method, to balance the weights between the two classes. The weights are calculated through the function compute_class_weight with class_weight = 'balanced' as parameter.

4.4. Training

Before training, a hyperparameter search was performed to optimize the performance of the selected models, respectively through *Grid Search* and *Randomized Search,* depending on the number of parameters involved.

4.4.1. Traditional Models

For traditional models, the dataset was initially split into *a training/ validation set* and *a test set,* using the Scikit -Learn train_test_split function . It was specified a stratified split via the stratify parameter to ensure that all sets contained the same distribution of samples for the positive class *(Hired* = 1) and the protected category *(Protected Category_encoded).* For some selected models it has been SelectKBest was also adopted , to identify the 15 most representative features in the dataset.

GridSearchCV and RandomizedSearchCV were used , depending on the number of parameter combinations for each model.

To perform the *cross validation,* StratifiedKFold was used on the *training/ validation set.*

The parameters submitted to *Gridsearch* for each traditional model, stored in a dictionary, are as follows:

- **Linear Regression:** fit_intercept, positive
- **Logistic Regression:** *C,* solver, class_weight, max_iter

- **Linear Perceptron:** early_stopping, class_weight
- **Gaussian Naïve Bayes:** var_smoothing
- **Decision Tree:** max_depth, class_weight
- **Random Forest:** max_depth, n_estimators, class_weight
- **AdaBoost:** n_estimators, learning_rate
- **K-Nearest Neighbors:** n_neighbors, weights, metric

The metrics used for the search are *f1_macro, recall_macro, precision_macro, roc_auc,* to ensure correct prediction of both classes. *Accuracy* was not used because it is not reliable in very unbalanced datasets, where it is sufficient to always predict the majority class to get a high score.

For each model, the one that best fits the four selected by the metrics was chosen. maximized the average.

The results obtained after training each model are represented by the *heatmap* in Figure 25.

Figure 25 Metric heatmap after Grid / Randomized Search

As expected, given that this is a very unbalanced dataset, most of the models tend to to predict the negative class more frequently, and for this reason it has high *accuracy ,* but relatively low *recall* compared to *precision.*

The parameters obtained for each model are shown in Figure 26.

```
DecisionTreeClassifier(max_depth=14, random_state=42)

LinearRegression()

LogisticRegression(C=100, class_weight='balanced', max_iter=500,
                   random_state=42)

GaussianNB(var_smoothing=1e-05)

RandomForestClassifier(max_depth=20, min_samples_leaf=2, n_estimators=200, random_state=42)

Perceptron(early_stopping=True, random_state=42)

KNeighborsClassifier(metric='manhattan', n_neighbors=2, weights='distance')

AdaBoostClassifier(learning_rate=1.5, n_estimators=40, random_state=42)
```

Figure 26 Parameters obtained for traditional models

4.4.2. Neural Network

The neural network training was performed on a balanced *dataset* with respect to the *Hired* target column through CTGAN, to optimize performance.

To maintain a *validation set* with only real candidates, not involved in the training of CTGAN, the *train/ validation set* has been explicitly split into *training set* and *validation set.* This choice allows to use the validation set in the *cross validation,* without risk of *data leakage* and the training set to generate synthetic samples.

The obtained synthetic samples were added to the original *training set* , to obtain a balanced dataset balanced_train_set with respect to the target.

Subsequently, the balanced_train_set was merged with the validation set to perform the *cross validation.*

For *cross-validation,* StratifiedKFold was used to define stratified *folds* of the validation test, composed only of real candidates. The *training folds* were then integrated with the samples of the balanced_train_set, thus increasing the number of samples available for training.

```
n_splits = 5
strat_kfold = StratifiedKFold(n_splits=n_splits, shuffle=True, random_state=random_state)

train_splits = []
val_splits = []

for train_index, val_index in strat_kfold.split(val_set, val_set['Hired']):
    train_index = train_index.tolist() + balanced_train_set.index.tolist()

    val_splits.append(val_index.tolist())

    train_splits.append(train_index)

cv_splits = [(train_splits[i], val_splits[i]) for i in range(n_splits)]
```

Figure 27 Using StratifiedKFold for ad hoc fold generation

Parameter Search

The selection of parameters was performed through RandomizedSearchCV, to reduce the computational cost and evaluate a greater number of configurations.

The parameters explored were:

- Number of neurons **per** layer (64, 128, 256, 512)
- **Activation function** (relu, elu, tanh)
- **Optimizer** (adam, rmsprop, sgd)
- **Batch size** (16, 32, 64, 128)
- **Number of epochs** (15, 30, 50, 100)
- **Learning rate** (0.001, 0.01, 0.1)

f1_weighted was selected as the metric for *Randomized Search,* to balance *precision* and *recall.*

The parameters obtained at the end, with an *f1_score* value higher than 0.98, are represented in Figure 28.

```
best_params_nn = {'optimizer': 'rmsprop',
                  'neurons': 256,
                  'learning_rate': 0.001,
                  'epochs': 50,
                  'batch_size': 64,
                  'activation': 'elu'}
```

Figure 28 Parameters selected by Randomized Search for the neural network

Model Evaluation

The best model got it with there *Randomized Search,* grid_result.best_estimator_, was used to make predictions on *the test set.*

In addition to the selected model, three additional neural networks were trained, using the optimal parameters previously obtained, but with different *seeds .*

The comparison between *training loss* and *validation loss* highlights a high variability in *validation set,* signaling a possible instability in the results.

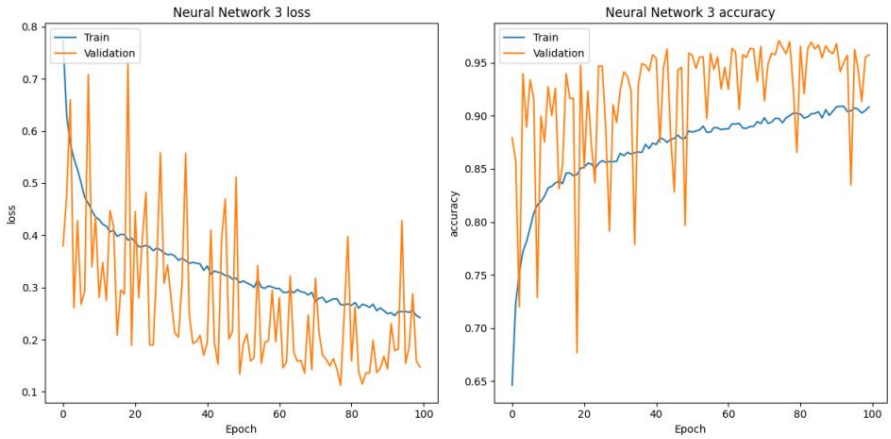

Figure 29 Train/ Validation Loss/ Accuracy Trends for the third neural network

However, the trained models show competitive performance, as shown in *heatmap* below (Figure 30):

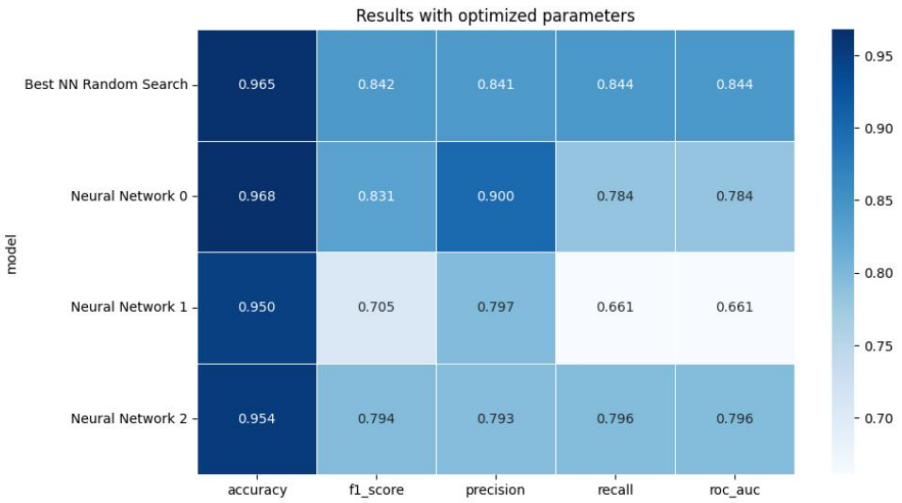

Figure 30 Performance of trained neural networks

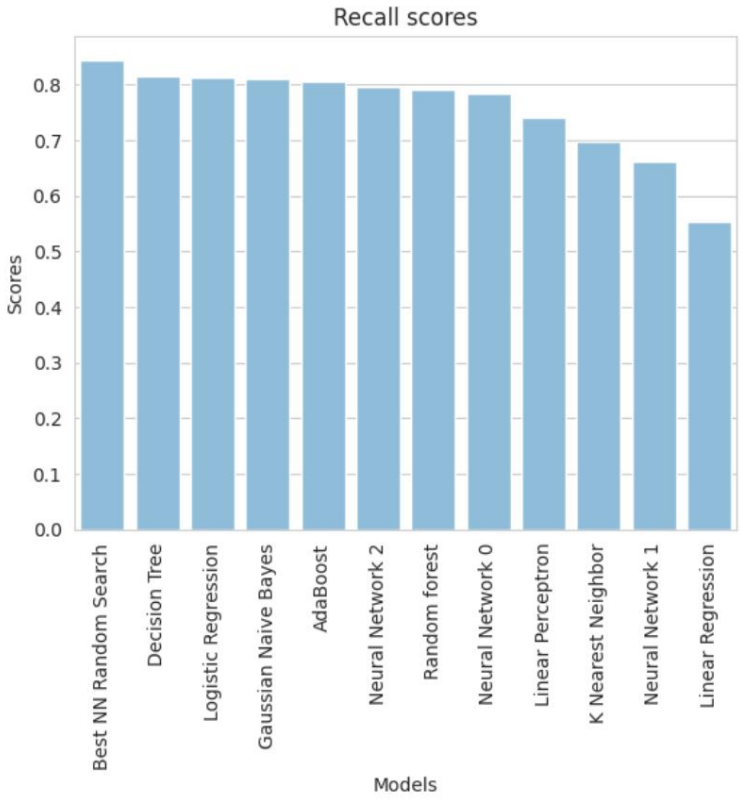

Figure 31 Recall histogram for each model

5. Bias Analysis

5.1.Selected metrics

To evaluate the level of *Fairness* of the trained models, three metrics were chosen:

- **Demographic Parity:** This metric requires that each group have the same opportunity to be assigned to the positive class *(Hired=1),* regardless of whether traits of a true or false positive. It could consider a model as *unfair* accurate, in the case of subgroups unbalanced towards the target column *Hired* in the *test set.* For example, in the reference dataset women have a higher rate of higher intake than men. Assuming that the test set has the same distribution, an accurate model could be considered *unfair* according to this metric. However, if the test set has imbalances for the *Hired* label caused by bias, these are identified if perpetuated by the model.

- **Equalized Odds:** This metric ensures that the *True Positive* and *False Odds* rates *Positive* are constant across different groups. This means, for example, that the model should incorrectly classify unsuitable candidates as positive with equal probability for individuals belonging to different categories, without favoring no subset.

- **Counterfactual Fairness:** This metric requires the model to re-perform the predictions on an alternative *test set ,* where only the value of one has been changed sensitive attribute. For example, if we change the gender of all candidates from *female* to *bad* the number of positive predictions increases, this means that with the same profiles the model favors samples with *Sex=male.*

5.2. Metrics Implementation

The three selected metrics were implemented through three ad hoc functions. It was also implemented an accessory function to manage *the output* of the latter and return

a *Pandas dataframe,* where each column represents a sensitive attribute and each row represents one of the implemented models.

5.2.1. Demographic Parity

The calculate_demographic_parity function implements the metric of the same name, checking for each subgroup of the specified sensitive attribute, the prediction rate positive.
For binary sensitive attributes, such as *Sex,* check that the difference between the two groups is below the specified tolerance threshold.

In the case of multi-class sensitive attributes, such as *Age Range,* use a chi-square test. square to check the dependence between predictions and sensitive attribute. In this case check that the *p-value* is above the *significance level threshold.*

If the model is found to be *unfair,* the function returns the subgroup that is found to be discriminated, which in this case it corresponds to the value with a lower percentage of positive predictions; alternative returns *True.*

As expected, the function, in Figure 32, does not take into account the actual labels of the test set y_test.

```
def calculate_demographic_parity(predictions, X_test, sensitive_feature, tolerance):

    df = pd.DataFrame({'predictions': predictions, 'sensitive_attribute': X_test[sensitive_feature]})
    positive_proportions = df.groupby(sensitive_feature)['predictions'].mean()
    percentage_difference = positive_proportions.max() - positive_proportions.min()

    if percentage_difference > tolerance:
        return positive_proportions.idxmin()
    return True
```

Figure 32 Pseudo code calculate_demographic_parity, binary case

5.2.2. Equalized Odds

The calculate_equalized_odds function calculates *True Positive Rate* (TPR) and *False Positive Rate* (FPR) for each subgroup identified by the sensitive attribute received among the

arguments. For both values check that the difference between the maximum and minimum value is below the tolerance threshold. If the difference exceeds tolerance the function returns the discriminated subgroup together with associated TPR and FPR, otherwise *True.*

In this case the discriminated subgroups are those for which the model predicted a lower rate of true positives or false positives.

```
def calculate_equalized_odds(predictions, true_labels, sensitive_feature, tolerance):

    metrics = {group: compute_TPR_FPR(predictions[sensitive_feature == group],
                                      true_labels[sensitive_feature == group])
               for group in unique(sensitive_feature)}

    max_tpr_diff = max(TPR_values(metrics)) - min(TPR_values(metrics))
    max_fpr_diff = max(FPR_values(metrics)) - min(FPR_values(metrics))

    if max_tpr_diff <= tolerance and max_fpr_diff <= tolerance
        return True
    return discriminated_groups
```

Figure 33 Pseudo code calculate_equalized_odds

5.2.3. Counterfactual Fairness

The calculate_counterfactual_fairness function implements the metric of the same name. In particular, for each subgroup identified by the specified sensitive attribute, create a dummy *test set* X_counterfactual in which it replaces all values of the sensitive attribute with the value of the subgroup under consideration. The new set is used to perform new predictions, through model.predict, compared with previous predictions.
For each subgroup, the consistency and the percentage difference are calculated.
positive delta predictions .

If the difference between delta_max and delta_min between the classes of the sensitive attribute is below the specified threshold tolerance and the detected consistency is greater than (1-tolerance) the function returns *True.*

Conversely, if the model turns out to be *unfair,* the discriminated groups are returned, together with the associated *delta* .

```
def calculate_counterfactual_fairness(predictions, model, sensitive_feature, X_test, tolerance):
    consistency = {}
    impact_delta = {}

    for group in unique(sensitive_feature):
        X_counterfactual = X_test.copy()
        X_counterfactual[sensitive_feature] = group

        new_preds = model.predict(X_counterfactual)
        new_preds = binarize(new_preds)

        consistency[group] = 1 - mean(abs(predictions - new_preds))
        impact_delta[group] = (sum(new_preds) - sum(predictions)) / len(predictions)

    if check_fairness(consistency, impact_delta, tolerance):
      return True
    return discriminated_group
```

Figure 34 Pseudo code calculate_counterfactual_fairness

5.3. Bias Detection

Demographic Parity

All trained models do not respect *Demographic Parity* (Figure 36) for *Age Range.*
This result could be due to the imbalance of the *Hired* class between the
Age Range subgroups .

The discriminated subgroup is identified by *Age Range = '<20 years',* which in the dataset provided has the lowest hiring rate, as seen in Figure 18.

Analyzing the distribution among the *Age Range* subcategories in the **test set ,** a
even more marked difference, as shown in Figure 35, with only 1.4% of the candidates
under 20, hired.

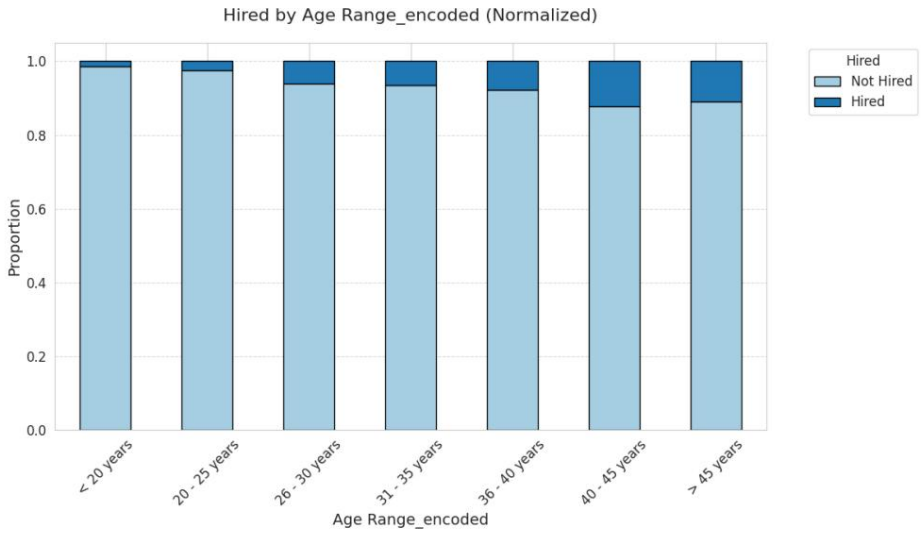

Figure 35 Hired distribution in the test set, by Age Range subgroups

These results can therefore be justified by the accuracy of the model, which predicts correctly account for disparities already present in the test set, which could be caused by biases or insufficient years of experience.

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree		< 20 years			
Linear Regression		< 20 years			
Logistic Regression		< 20 years		non-european	
Gaussian Naive Bayes		< 20 years			
Random forest		< 20 years			
Linear Perceptron		< 20 years	no	non-european	
K Nearest Neighbor		< 20 years			
AdaBoost		20 - 25 years			
Neural Network 0		< 20 years			
Neural Network 1		> 45 years			
Neural Network 2		< 20 years			
Best NN Random Search		20 - 25 years			

Figure 36 Demographic Parity

From the analysis conducted it emerges that this metric "forces" the models to balance protected categories, even in the case of *biased test sets.* However, this results in higher performance low, necessary to avoid reflecting the discrimination already present.

5.3.1. Equalized Odds

As mentioned, this metric is used to verify the difference between *true* and *false positives. rate* of different subgroups does not exceed a set threshold tolerance.

As we can see from the results obtained, in Figure 37, most of the models tend to to have a very unbalanced true positive rate. The subgroups that present the minimum TPR are those for which the model is less accurate in predicting the positive class.

However, investigating the distribution of the subgroups in the *test set* it emerges that both *Protected Category = yes* and for *European Residence = no,* there is only one sample hired *(Hired = 1).* This means that TPR for these groups can only assume 0 or 1. For this reason any other subgroup will typically have a TPR greater than 0 or less than 1.

The case of *Italian Residence* is similar, with only two non-Italian candidates hired.

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree		< 20 years [TPR=0.5]	no [TPR=0.0]	non-european [TPR=0.0]	no [TPR=0.643]
Linear Regression		< 20 years [TPR=0.0]			
Logistic Regression		26 - 30 years [TPR=0.75]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Gaussian Naive Bayes		26 - 30 years [TPR=0.625]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Random forest		40 - 45 years [TPR=0.375]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Linear Perceptron		< 20 years [TPR=0.5]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
K Nearest Neighbor		40 - 45 years [TPR=0.125]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
AdaBoost		40 - 45 years [TPR=0.5]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Neural Network 0		26 - 30 years [TPR=0.5]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Neural Network 1		> 45 years [TPR=0.176]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Neural Network 2		20 - 25 years [TPR=0.5]	no [TPR=0.0]	non-european [TPR=0.0]	no [TPR=0.612]
Best NN Random Search		20 - 25 years [TPR=0.5]	no [TPR=0.0]	non-european [TPR=0.0]	no [TPR=0.653]

Figure 37 Equalized Odds

5.3.2. Counterfactual Fairness

The results of this metric, in Figure 38, highlight the difference for each subgroup percentage of the number of positive predictions.

For example, Neural Network *2,* for a dummy test set in which all candidates have as *Sex male,* predicts about 0.2% fewer positive labels. The same network seems instead be more favorable towards women, with 2.5% more positive predictions, highlighting a possible bias within the *Sex category.*

Some models seem to discriminate against Italian candidates, mainly because favor non-Italian candidates. These results will be analyzed in the next phase through ad hoc *explainers .*

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree					
Linear Regression					
Logistic Regression		< 20 years [-4.93%], > 45 years [8.78%]			
Gaussian Naive Bayes					
Random forest					
Linear Perceptron					
K Nearest Neighbor					
AdaBoost					
Neural Network 0			yes [-0.06%], no [6.05%]		
Neural Network 1			yes [-0.18%], no [19.53%]		
Neural Network 2	male [-0.24%], female [2.55%]	< 20 years [-1.42%], > 45 years [2.67%]	yes [-0.06%], no [15.25%]	european [0.24%], non-european [0.71%]	yes [-1.66%], no [0.0%]
Best NN Random Search			yes [0.18%], no [13.83%]	european [0.06%], non-european [13.59%]	

Figure 38 Counterfactual Fairness

5.4. Explainability

AI explainability is essential to better understand the decision-making process of models, improving transparency and end-user trust.

In this context two *explainers* were used , LIME *(Local Interpretable Model-agnostic Explanations)* and SHAP *(SHapley Additive exPlanations)* to analyze attributes most influential for each model. Through these tools it is possible to verify the influence of sensitive attributes on models and the impact of particular values on results.

Furthermore, by analyzing the most important attributes it is possible to identify discrimination indirect in the models.

5.4.1. LIME

LIME generates local explanations for *black-box models,* through input generation perturbed to analyze the behavior of the model. In this way it is able to

distinguish which attributes most influenced the prediction for a specific sample and in what direction.[18]

The 20 most influential features for each cluster of the test set identified by a specific sensitive attribute.

Sex

As anticipated by the metrics used, no model presents the sensitive attribute *Sex* among the most influential attributes. This result is due to the dataset used, which does not present particular imbalances for the *Hired* target column between different genders.

However, all the analyzed models are strongly influenced by the fields related to RAL: *Ral Maximum, Expected Ral, Minimum Ral.*

As seen in paragraph 4.2.3, these columns are most likely subject to bias and their influence could therefore introduce **indirect discrimination.**

Protected Category

For candidates belonging to protected categories, it emerged that no model uses, among the 20 Most Influential Features, the sensitive attribute *Protected Category_encoded.*

This allows us to exclude any possible direct discrimination against this attribute. However, even in this case, the results show a strong influence by the RAL related fields.

[18] Alikhademi et al., «Can Explainable AI Explain Unfairness?»

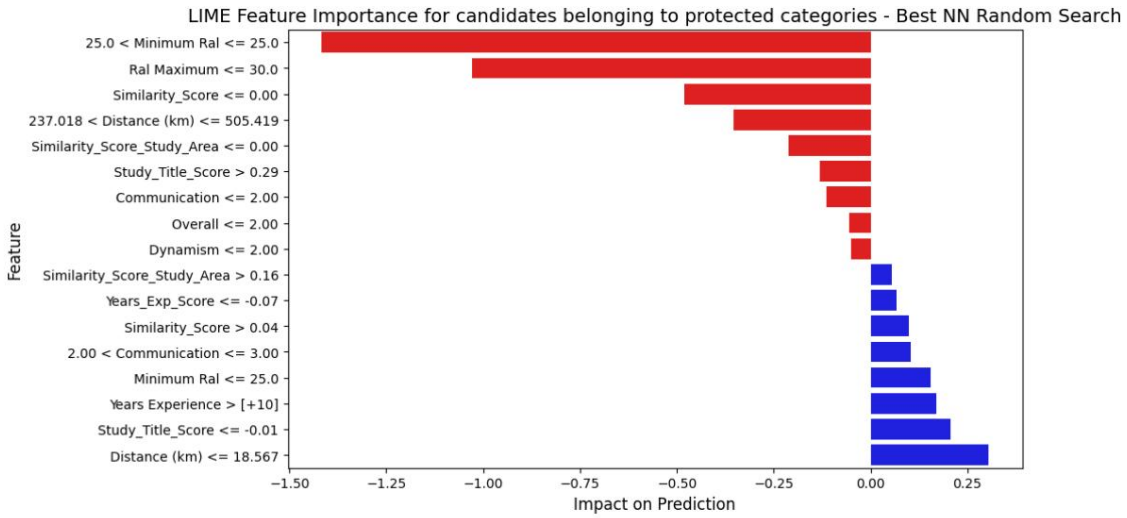

Figure 39 Influence distribution of the 20 most important features for the neural network model, for candidates belonging to protected categories

Age Range

Analyzing the behavior of models for candidates under the age of twenty, it emerges that most of the models are influenced by the sensitive attribute *Age Range,* highlighting a possible **direct discrimination** against very young candidates young people. This result was also highlighted through the fairness metrics used in the previous section.

Here too, the RAL columns are very influential.

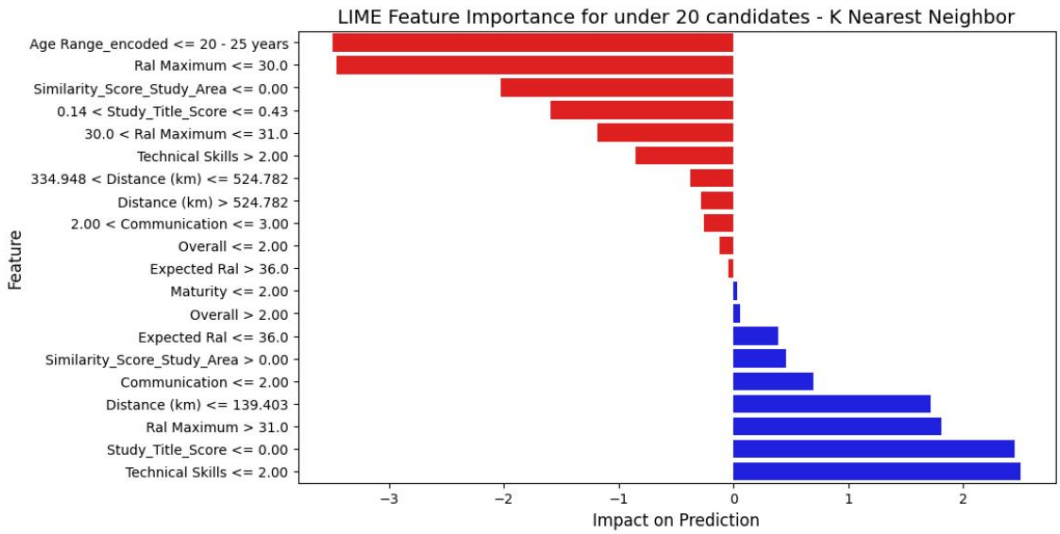

Figure 40 Influence distribution of the 20 most important features for the K Nearest Neighbor model, for candidates with Age Range = '< 20 years'

Italian Residence

For candidates not resident in Italy, no model shows a particular influence from part of the sensitive attribute *Italian Residence.*

However, most of the models show attributes related to it, such as *Distance (km), Residence Country, Residence Italian Region, Residence Italian Province.*

However, it is not possible to condemn the use of distance from the workplace since it is often essential for choosing a candidate. Even in this case the models are highly influenced by the RAL-related fields.

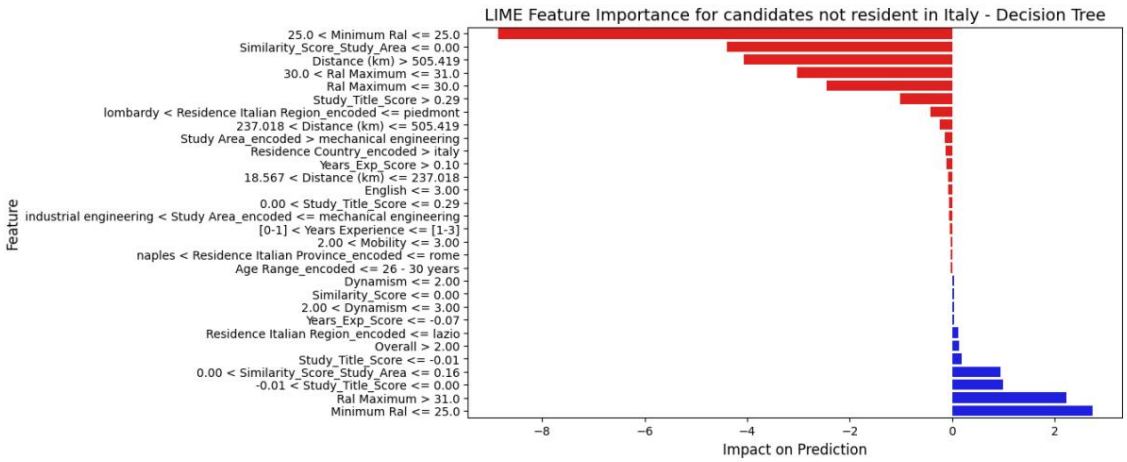

Figure 41 Influence distribution of the 20 most important features for the Decision Tree model, for candidates with Italian Residence = 'no'

European Residence

Among the implemented models only Decision Tree highlights the use of the sensitive attribute *European Residence,* with negative influence, as shown in Figure 42.

Again, all models are influenced by related fields, such as *Distance (km)* and those relating to the *RAL.*

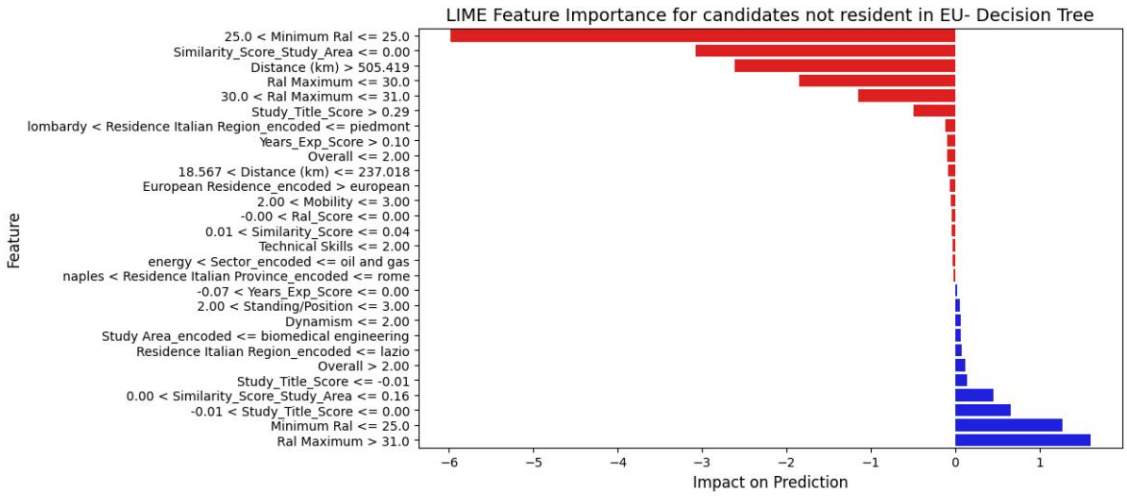

Figure 42 Influence distribution of the 20 most important features for the Decision Tree model, for candidates with European Residence = 'non-European'

5.4.2. SHAP

The SHAP explainer attributes the change in the model's prediction to each feature.
when conditioned, thus revealing the extent to which each characteristic contributes to the classification, both positively and negatively.19

Even with SHAP, the most relevant attributes are those related to RAL.

Analyzing the results of Neural Network 2, which had obtained questionable results for *Counterfactual Fairness* (Figure 38), the use of different sensitive attributes emerges. In particular, analyzing Figure 43, it is possible to notice among the attributes:

- *Age Range:* High values of *Age Range* positively influence the result. Also the related attribute *Years Experience* follows the same trend.

- *Italian Residence:* Low values of the corresponding *Italian* column *Residence_encoded* positively influence the prediction. This column is binary and 0 identifies candidates with non-Italian residence. This result is consistent with the results obtained from the metrics.

[19] Goethals, Martens, and Calders, «PreCoF».

- *European Residence:* For *European Residence* the negative impact of values is evident high, which identify non-European candidates.

- *Protected Category:* Again *Protected Category_encoded = 1,* which identifies candidates belonging to a protected category, negatively influences the result.

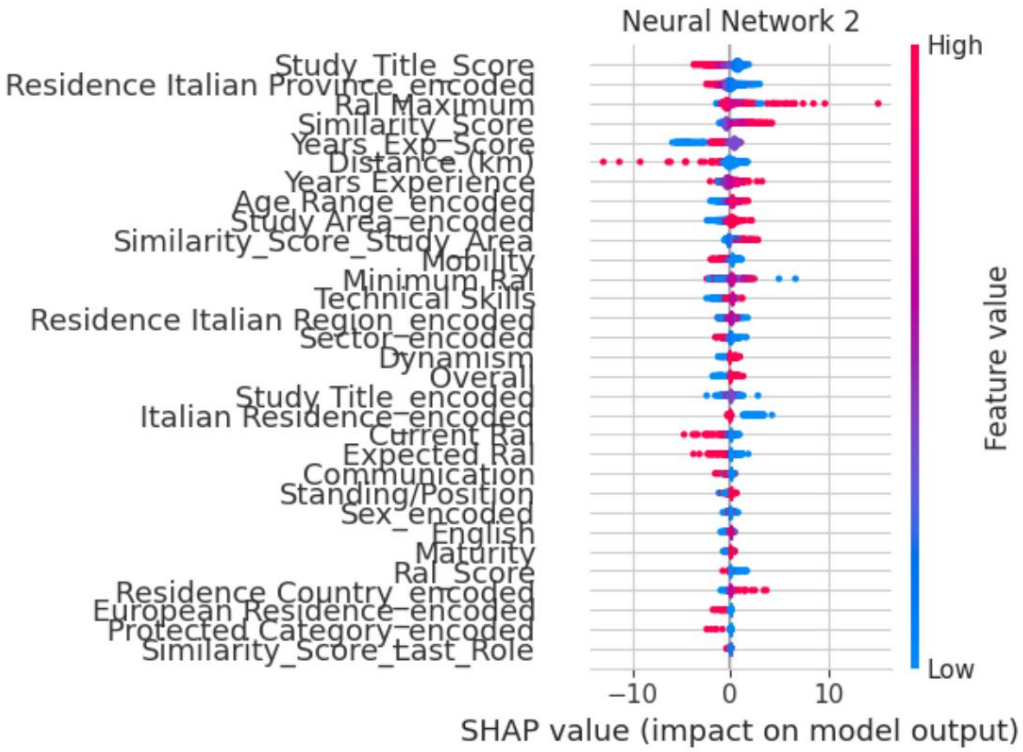

Figure 43 SHAP explanations for Neural Network 2

6. Bias Mitigation

6.1. Pre-processing mitigation

The analysis of the dataset did not reveal any systematic bias towards protected categories. As discussed in Section 4.2.3, in the case of the *Sex* attribute , women have a higher rate of slightly higher intake than men. This slight imbalance
It could be due to a greater competence of the candidates (Figure 11), although there is no certain data to confirm it. Akkodis recruiters also stated that there is no compensation process within the company towards the candidate.

In the case of other sensitive attributes, such as *Protected Category, Age Range* or *Italian Residence,* the slight imbalance in the rate of candidates hired is mainly due to:

- **Under-representation** of some groups in the dataset;
- **Reasonable factors,** such as distance from the workplace or lack of years of experience needed.

To facilitate a better evaluation of the *fairness* metrics, the *dataset* has been divided entrusting the *test set* with a larger portion than in the previous phase, equal to 40% of the total.

Subsequently, all attributes that showed bias during the dataset analysis.

Finally, *oversampling techniques,* through CTGAN, and *undersampling,* were used to balance underrepresented classes in the *training set.*

6.1.1. Removing Biased Attributes

Although the dataset provided has shown greater fairness than others traditionally used in this context, the fields relating to *RAL* have highlighted hidden critical issues.

As noted in Section 4.2.3, the RAL expected from the job position associated with each candidate hired appears to be influenced by the sensitive attributes. In fact, there is a disparity between candidates with the same educational qualification or years of experience. belonging to different groups.

For these considerations all the fields relating to *RAL* have been removed, to avoid a possible indirect discrimination based on the latter.

6.1.2. Data Augmentation

The *training set* was expanded, to compensate for its resulting reduced size of the different subdivision of the dataset, and balanced through synthetic samples, generated via **CTGAN** *(Conditional Tabular GAN),* a model based on generative adversarial networks designed for tabular data. 20 CTGAN was chosen for its ability to represent compared to other tools and for its native support for categorical variables.

The adopted strategy involves training a separate CTGAN model for each subgroup, identified by a sensitive attribute and the target column *Hired.* This has allowed to maintain internal relationships within subgroups, avoiding distortions in the data synthetics.

For each trained CTGAN model , synthetic samples were generated until reaching at least 40% of the size of the majority class, relating to the same attribute sensitive, ensuring a more balanced presence for each subgroup.

Finally, all samples were concatenated to the original *training set ,* bringing the total of the candidates over 24,000.

After oversampling, new models were trained, keeping the same ones hyperparameters previously optimized with *Grid Search.*

[20] Panagiotou, Roy, and Ntoutsi, «Synthetic Tabular Data Generation for Class Imbalance and Fairness».

Results

As expected, oversampling had a moderately negative impact on performance.
of the models (Figure 44), with the exception of K Nearest Neighbor.

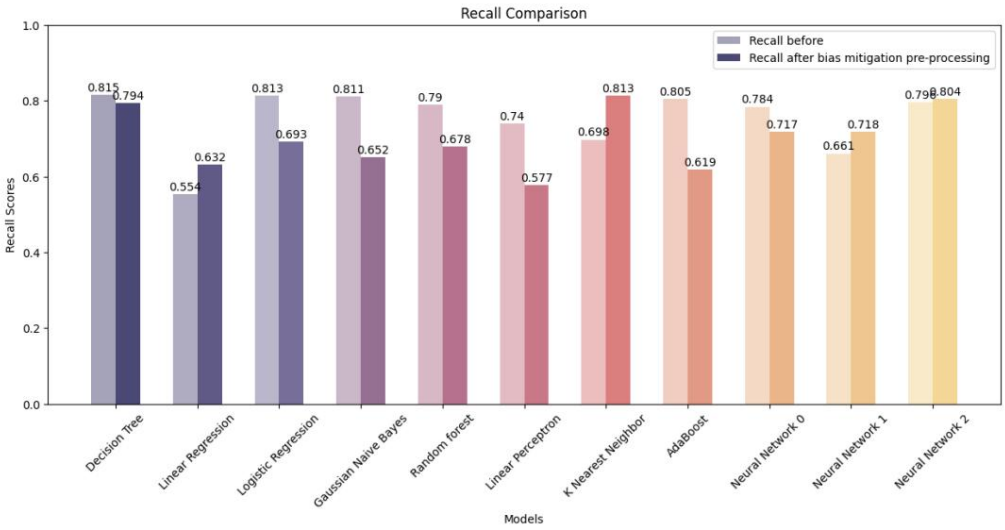

Figure 44 Recall Score comparison before and after removing RAL columns and oversampling with CTGAN

However, it did not lead to a noticeable improvement in *fairness.* This result is mainly due to the very small size of the initial *dataset .* Despite the *test set* now represents 40% of the candidates, the data is still insufficient to allow for a stable improvement in metrics, which instead oscillate between two extremes.

The underrepresentation of protected categories in the *test set* amplifies the impact of any variation in predictions, causing extreme swings in the direction of the bias.

In the case of **Demographic Parity,** in Figure 45, it is possible to observe a clear change in the direction of bias, with models that appear to favor candidates belonging to protected category. However, this apparent imbalance is due to the limited number of candidates of this type within the test set. The rate of positive predictions in fact increases very quickly for these candidates, amplifying the effect of positive predictions additional.

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree		< 20 years			
Linear Regression	male				
Logistic Regression	male	< 20 years	yes		
Gaussian Naive Bayes	male				no
Random forest					
Linear Perceptron					
K Nearest Neighbor		< 20 years			
AdaBoost					
Neural Network 0		< 20 years			no
Neural Network 1		< 20 years			
Neural Network 2	male	< 20 years	no	non-european	

Figure 45 Demographic Parity of models trained on the oversampled training set, without RAL fields

Even in the case of **Equalized Odds** an improvement is observed: some biases have been resolved while others have been improved if not even reversed.

For the Gaussian Naïve Bayes model it can be noted that the candidates belonging to protected category had a higher rate of false positives, highlighting a change of model in this direction. Again, the false positive rate is increasing rapidly, as the unhired candidates belonging to protected categories are only 18.

In the case of *Italian Residence* it is possible to observe that most of the models have solved the bias while the rest have reversed it towards the majority class *(Italian Residence = yes),* also in this case highlighting an improvement, albeit sudden, in this direction.

The change in the *Sex* attribute is instead due to the removal of the columns concerning the *RAL,* which showed a strong imbalance towards women. By removing attributes that are clearly discriminatory towards female candidates and keeping in mind that the starting dataset already predicted a higher rate of hiring For women, the metrics show a shift in the opposite direction.

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree		20 - 25 years [TPR=0.444]		non-european [TPR=0.0]	yes [TPR=0.0]
Linear Regression	male [TPR=0.218]	> 45 years [TPR=0.192]			
Logistic Regression	male [TPR=0.489/FPR=0.148]	36 - 40 years [TPR=0.412]			no [TPR=0.607]
Gaussian Naive Bayes	male [TPR=0.286]	> 45 years [TPR=0.192]			no [TPR=0.362/FPR=0.063]
Random forest					yes [TPR=0.0]
Linear Perceptron		36 - 40 years [TPR=0.059]		european [TPR=0.189]	no [TPR=0.189]
K Nearest Neighbor			yes [TPR=0.644]	european [TPR=0.648]	yes [TPR=0.0]
AdaBoost			no [TPR=0.0]	non-european [TPR=0.0]	no [TPR=0.255]
Neural Network 0	male [TPR=0.391]	< 20 years [TPR=0.353]	yes [TPR=0.454]		
Neural Network 1		< 20 years [TPR=0.294]	yes [TPR=0.459]		
Neural Network 2		< 20 years [TPR=0.412]	no [TPR=0.5]	non-european [TPR=0.0]	yes [TPR=0.5]

Figure 46 Equalized Odds of models trained on the oversampled training set, without RAL fields

From the analysis of **Counterfactual Fairness** the new direction of the models, which resolved or reversed the direction of the bias.

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree					
Linear Regression	male [-2.34%], female [10.45%]				
Logistic Regression	male [-8.49%], female [28.75%]	< 20 years [-4.45%], > 45 years [7.92%]			yes [-10.77%], no [0.06%]
Gaussian Naive Bayes	male [-2.31%], female [8.81%]				no [-0.12%], yes [25.67%]
Random forest					
Linear Perceptron	male [-1.81%], female [8.31%]				
K Nearest Neighbor		< 20 years [0.42%], > 45 years [17.83%]			
AdaBoost					
Neural Network 0		< 20 years [-0.83%], > 45 years [9.38%]	yes [0.68%], no [29.64%]	european [0.18%], non-european [36.02%]	
Neural Network 1		< 20 years [-2.37%], > 45 years [10.12%]	yes [0.5%], no [22.26%]	european [0.15%], non-european [27.18%]	
Neural Network 2		< 20 years [-5.1%], > 45 years [21.69%]	yes [0.71%], no [29.64%]	european [0.21%], non-european [37.72%]	yes [-16.2%], no [0.09%]

Figure 47 Counterfactual Fairness of models trained on the oversampled training set, without RAL related fields

6.1.3. Dataset Balancing

After *oversampling,* a further balancing of the *dataset* was performed through undersampling, to better balance and mitigate any distortions introduced.
This controlled sub-sampling technique allows to limit the impact of the classes most represented.

The method used assigns to each sample a weight inversely proportional to the size of the group to which they belong, allowing for balanced sampling, without alter the overall size of the *training set.* During the selection, favors were given real candidates.

Finally, the new *training set* was used to train a new set of models.

Results

As expected, undersampling had a further negative impact on the performance of the models, as shown in Figure 48, where the recall scores were compared
obtained from the models in the three cases seen.

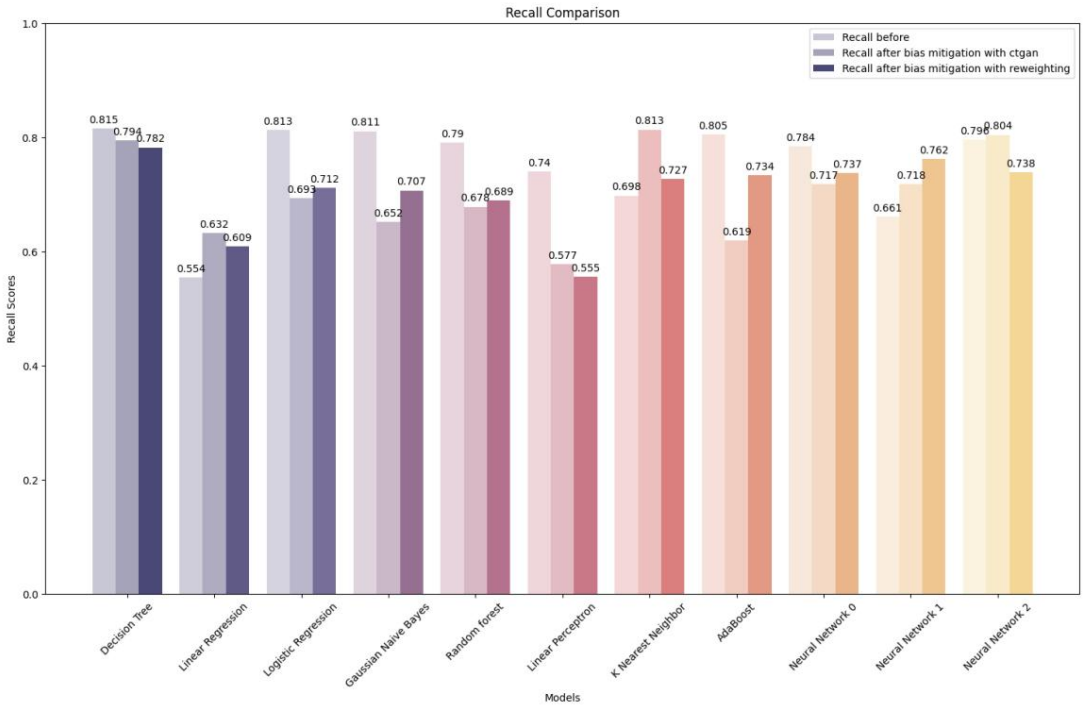

Figure 48 Recall Score comparison before and after removing RAL columns and oversampling with CTGAN

However, compared to the previous phase, no particular positive variations are observed in the fairness metrics , which show only slight changes.

For this reason, in the next phase, the over-sampled *training set* will be used.
via CTGAN.

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree		< 20 years			
Linear Regression					
Logistic Regression	male	< 20 years	yes		yes
Gaussian Naive Bayes		< 20 years	yes	european	yes
Random forest		< 20 years			
Linear Perceptron		< 20 years			
K Nearest Neighbor		20 - 25 years			
AdaBoost		< 20 years			
Neural Network 0		< 20 years			
Neural Network 1		< 20 years			
Neural Network 2		< 20 years			

Figure 49 Demographic Parity of models trained on the undersampled training set

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree		40 - 45 years [TPR=0.462]			yes [TPR=0.0]
Linear Regression		> 45 years [TPR=0.077]		european [TPR=0.24]	yes [TPR=0.0]
Logistic Regression	male [TPR=0.624/FPR=0.234]	> 45 years [TPR=0.538], < 20 years [FPR=0.233]		non-european [TPR=0.5]	yes [TPR=0.0/FPR=0.056]
Gaussian Naive Bayes		> 45 years [TPR=0.385]	yes [FPR=0.15]	european [TPR=0.571/FPR=0.152]	yes [TPR=0.0]
Random forest		26 - 30 years [TPR=0.275]			yes [TPR=0.0]
Linear Perceptron		> 45 years [TPR=0.038]		european [TPR=0.143]	
K Nearest Neighbor		26 - 30 years [TPR=0.375]	yes [TPR=0.495]		yes [TPR=0.0]
AdaBoost		> 45 years [TPR=0.346]	no [TPR=0.0]	non-european [TPR=0.0]	yes [TPR=0.0]
Neural Network 0		< 20 years [TPR=0.294]	yes [TPR=0.495]		
Neural Network 1		< 20 years [TPR=0.294]			
Neural Network 2		26 - 30 years [TPR=0.362]	yes [TPR=0.5]		

Figure 50 Equalized Odds of the models trained on the undersampled training set

	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Decision Tree					
Linear Regression					
Logistic Regression	male [-5.04%], female [17.66%]	< 20 years [-5.85%], > 45 years [9.26%]			yes [-28.66%], no [0.27%]
Gaussian Naive Bayes					yes [-18.61%], no [0.21%]
Random forest					
Linear Perceptron					
K Nearest Neighbor		< 20 years [-0.12%], > 45 years [13.09%]			
AdaBoost					
Neural Network 0		< 20 years [-1.34%], > 45 years [13.26%]	yes [0.77%], no [23.83%]	european [0.0%], non-european [33.77%]	
Neural Network 1		< 20 years [-1.93%], > 45 years [21.66%]	yes [0.98%], no [39.23%]	european [0.18%], non-european [43.65%]	
Neural Network 2		< 20 years [-1.42%], > 45 years [11.36%]	yes [0.77%], no [19.53%]	european [0.06%], non-european [35.7%]	

Figure 51 Counterfactual Fairness of models trained on the undersampled training set

6.2. In-process mitigation: Adversarial Debiasing

Adversarial Debiasing is a technique based on adversarial neural networks, used to mitigate *biases* in a model during training. The implementation includes the placement of adversary models alongside the main model *(Main Model) Models),* which aim to predict a specific sensitive attribute, using the *Main Model* 's predictions as input . If the adversarial networks perform accurate predictions means that the main model is particularly influenced by the sensitive attributes in his predictions.

Training takes place alternately between the main model and the enemy models. In this case the neural network does not simply minimize its own loss function, but optimizes a combined *loss ,* which includes the difference between its loss function and that of the opponent models. The main model's weight update is also directed by the combined loss function. This pushes the main model to find representations that are predictive for its target but not for sensitive attributes, thus reducing *bias.*

If the enemy models have high accuracy, it means that the *Main Model* is still exploiting the sensitive attributes in his decisions. Conversely, if the opponents' *loss* increase the main model has learned a more equitable representation.

6.2.1. Architecture

- ***Main Model:*** The *Main Model* has the same structure as the neural networks used so far, composed of three *hidden layers, batch normalization* and *dropout.* The parameters used coincide with those optimized through *Grid Search* in the previous phases.

- ***Adversary Models:*** The structure of adversary networks is simpler and consists of two *hidden layers* and *batch normalization.* Depending on the associated sensitive attribute *The output layer* is either single neuron, for binary *features* like *Sex,* or multiple neuron with *softmax activation.* The loss function of each adversarial model is weighted by the class distribution of the sensitive attribute, to avoid that the majority class dominates the training.

6.2.2. Equalized Odds Penalty

A penalty function based on *Equalized Odds has also been implemented,* integrated in the overall loss function. The latter measures the differences in error rates between sensitive groups and pushes the model to make fairer predictions between the different subgroups.

```
def equalized_odds_penalty(y_true, y_pred, sensitive_group):

    for group in groups:
        tn, fp, fn, tp = confusion_matrix(group['true_labels'], group['predictions'], labels=[0, 1])
        tpr = tp / (tp + fn) if tp + fn != 0 else 0
        fpr = fp / (fp + tn) if fp + tn != 0 else 0

    max_tpr_diff = max(tprs) - min(tprs)
    max_fpr_diff = max(fprs) - min(fprs)

    penalty = max(max_tpr_diff, max_fpr_diff)
    return penalty
```

Figure 52 Pseudo code for Equalized Odds Penalty

6.2.3. Training

The training was performed using the *training set* obtained by *oversampling* with CTGAN, to combine different *bias mitigation* techniques .

The *Main Model* was trained sequentially with each of the five adversarial networks, relating to sensitive attributes: *Sex, Age Range, Italian Residence, European Residence* and *Protected Category.*

Results

The model's performance is lower than that of models not subjected to correction techniques. *bias mitigation.* Analyzing the scores obtained by the main network, its tendency emerges in negatively labeling a good part of the samples.

This more cautious behavior is due to competition with opposing networks.
Since the *test set* is biased towards the positive label, the *accuracy* of the model it remains high anyway.

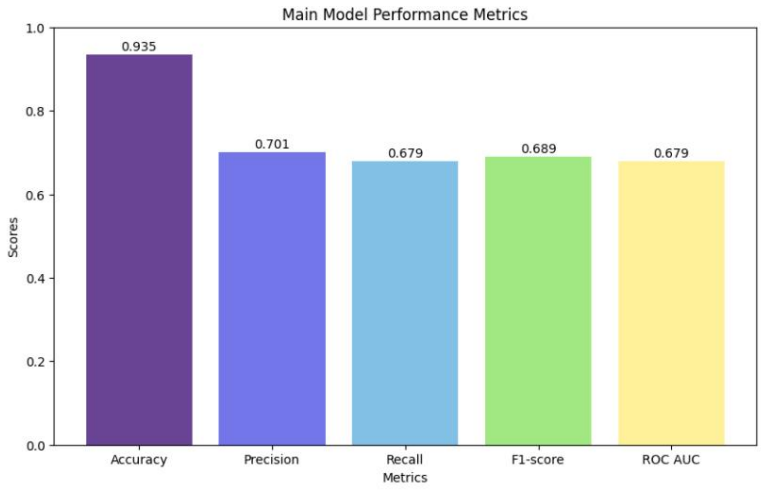

Figure 53 Performance Main Model Adversarial Debiasing

However, fairness metrics show promising improvements. Again, *Counterfactual Fairness,* in Figure 54, highlights a change of direction by the model, even if it needs to be corrected.

Metric	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Demographic Parity		< 20 years			
Equalized Odds		< 20 years [TPR=0.118]		non-european [TPR=0.0]	
Counterfactual Fairness	male [-0.15%], female [0.92%]	< 20 years [-1.66%], > 45 years [15.1%]	yes [0.5%], no [20.42%]	european [0.09%], non-european [28.96%]	

Figure 54 Metrics obtained for the Main Model after Adversarial Debiasing

Explainability

The graph obtained with SHAP highlights a strong dependence on the province *(Residence Italian Province)* and region of residence *(Residence Italian Region).* This This result is due both to the low dimensionality of the *test set,* which is not very representative, both to the fact that most of the job positions managed are concentrated in the city specific, such as Milan, Turin, etc. (Figure 56).

It is quite reasonable that candidates residing in the same city are at an advantage.

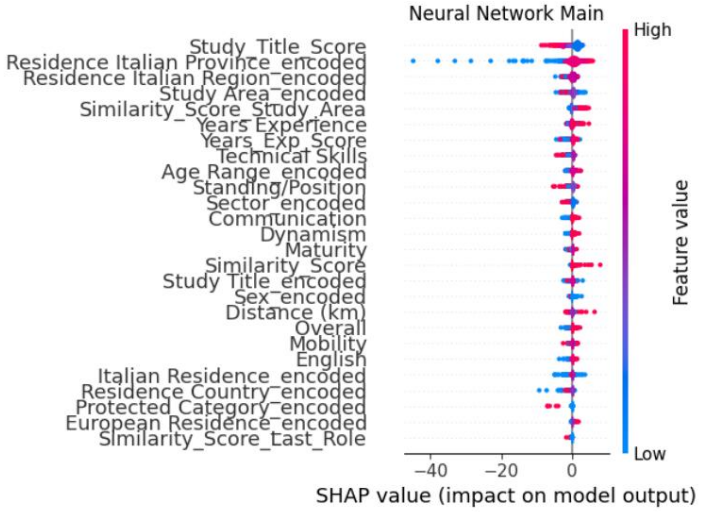

Figure 55 Shap Explainer for the Main Model

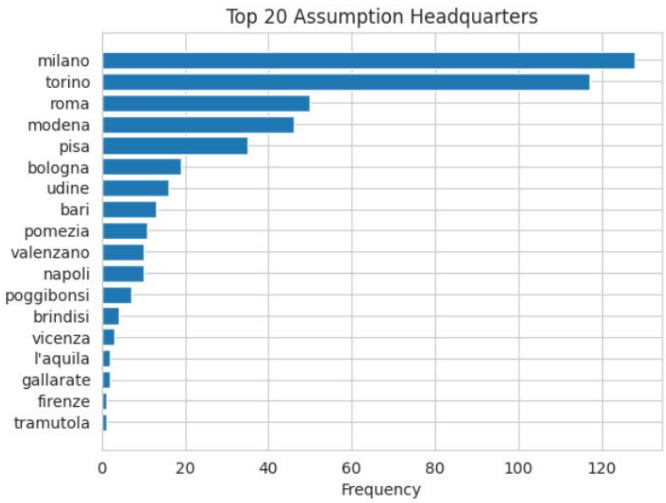

Figure 56 Distribution of the 20 most frequent workplaces present in the dataset

6.2.4. Optimal Threshold Selection

To improve the overall performance, an optimal threshold was calculated, through the *ROC Curve* (Figure 57), to calculate the point of greatest balance between TPR and FPR.

Using the threshold obtained in this phase the overall performance of the model improve, while its accuracy worsens, as it classifies more positively candidates.

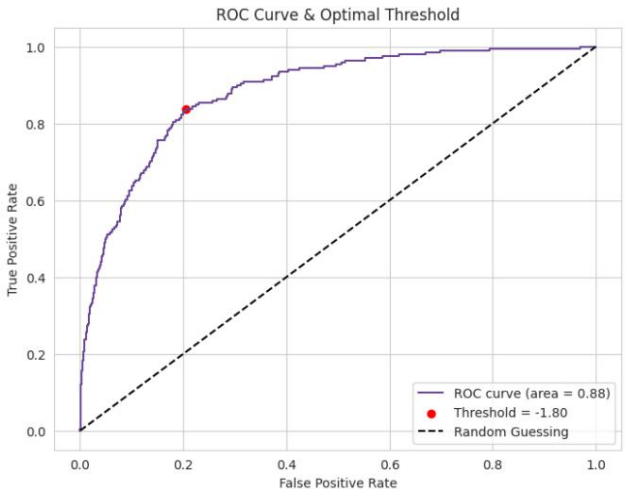

Figure 57 ROC curve for Main Model

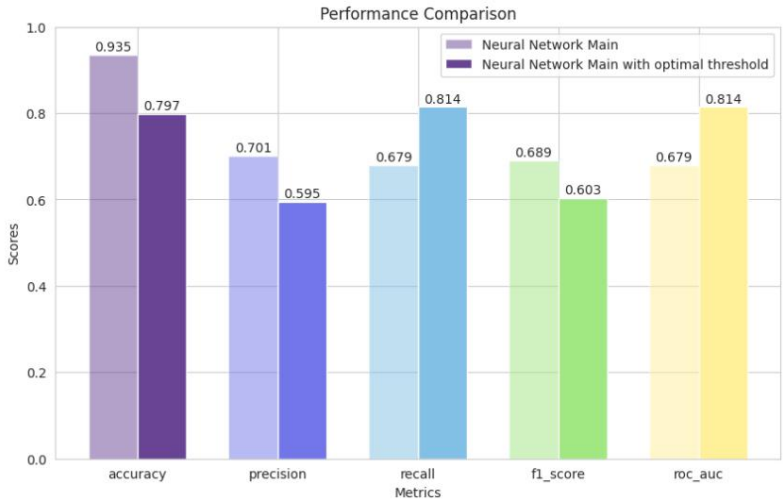

Figure 58 Comparison of the performance of the Main Model before and after using the optimal threshold

However, using a single threshold may not be the best choice in terms of *fairness.*

The new metrics calculated after applying the threshold are shown in Figure

59. The results highlight that the model still favors some categories.

	Metric	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Neural Network Main	Demographic Parity	male	20 - 25 years	no	non-european	
Neural Network Main	Equalized Odds		< 20 years [TPR=0.647]		non-european [TPR=0.5]	yes [TPR=0.5]
Neural Network Main	Counterfactual Fairness	male [-19.17%], female [-18.1%]	< 20 years [-20.68%], > 45 years [-3.92%]	yes [-18.52%], no [1.39%]	european [-18.93%], non-european [9.94%]	yes [-24.07%], no [-19.08%]

Figure 59 Main Model Metrics with Optimal Threshold

6.3. Post-processing mitigation: Fair Thresholding

In machine learning models , using a fixed threshold for classification can introduce *bias,* penalizing some groups more than others. In the phase preceding the use of the threshold obtained via *ROC Curve* led to a disparity in model performance against different groups, as shown by the metrics in Figure 59.

To resolve this imbalance, thresholds were calculated in post-processing. differentiated for each subgroup, optimized according to the respective *ROC Curve.* This has allowed to reduce classification errors at local level, improving the balance between sensitivity and specificity of the model for each group.

Using custom thresholds shows performance improvement, such as shown in Figure 60.

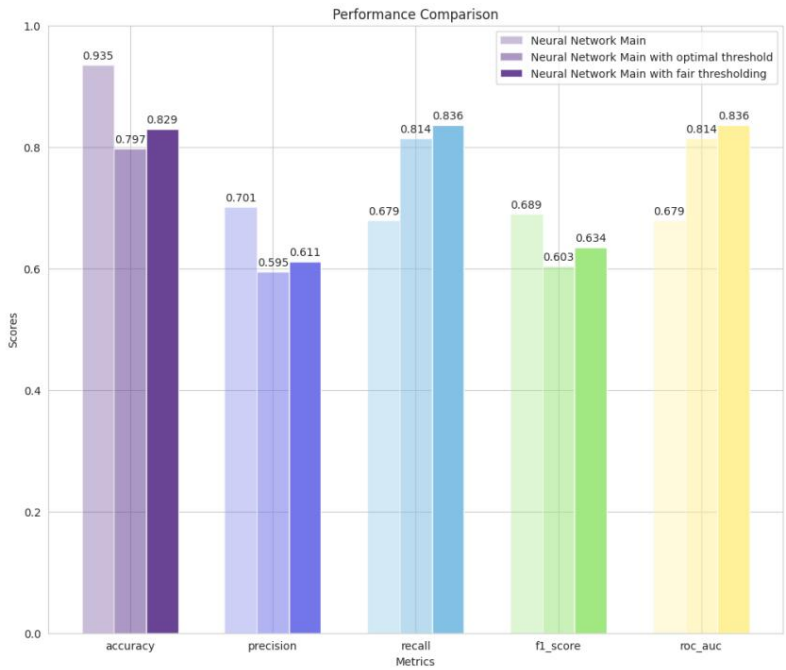

Figure 60 Performance comparison of the Main Model depending on the type of threshold used

The *fairness* metrics obtained after applying the custom thresholds are improved, as illustrated in Figure 61.

According to **Demographic Parity,** the network no longer discriminates against candidates belonging to the category *bad* while the predictions in the case of the *Age Range* attribute are now unbalanced in the comparisons of the *36-40 age group,* which represents the group with the highest rate of assumption in the dataset (Figure 18).

Equalized Odds no longer highlights critical issues thanks to the use of optimized thresholds with the ROC curves.

In the case of **Counterfactual Fairness,** an improvement is observed, with a reduction in the disparity between classes of the same sensitive attribute.

	Metric	Sex_encoded	Age Range_encoded	Italian Residence_encoded	European Residence_encoded	Protected Category_encoded
Neural Network Main	Demographic Parity		36 - 40 years	no	non-european	
Neural Network Main	Equalized Odds					
Neural Network Main	Counterfactual Fairness	male [-16.08%], female [-15.01%]	< 20 years [-17.6%], > 45 years [-0.83%]	yes [-15.43%], no [4.48%]	european [-15.85%], non-european [13.03%]	yes [-20.98%], no [-15.99%]

Figure 61 Fairness metrics after applying different thresholds for each subgroup

7. Discussion

The analysis conducted on the dataset provided by Akkodis has highlighted how important it is consider *fairness* in developing machine learning models , especially in contexts critics like the one treated by the company. Equity is not only an ethical issue, but it promotes reliable decisions free from systematic distortions.

This study focused on identifying and assessing *biases* present in the *dataset* and subsequently acquired by the models developed with it. The identification is occurred through selected metrics, capable of providing a vision of the equity to be different perspectives. The level of *fairness* of a model cannot be assessed with a single metric but requires an in-depth and multidisciplinary analysis, evaluating in addition to the overall performance also the distribution of errors in the individual groups.

To mitigate the disparities found, several mitigation strategies were employed throughout the model development process. In the *pre-processing phase,* the dataset was balanced to reduce the most obvious *biases* and compensate for the underrepresentation of some classes. During the training of the model, the *Adversarial* technique was also used *Debiasing,* aimed at reducing the use of sensitive attributes by the network. Finally, in *post-processing, ad hoc* thresholds were developed for each subgroup of the dataset, optimized via *ROC Curves,* to ensure good local performance.

The level of fairness was monitored throughout the development, to evaluate *in real-time* the impact of each applied technique. However, the main limitation of this thesis is given by the small size of the dataset, which made it more difficult to highlight the full potential of the mitigation techniques applied, in addition to the original structure which presented positions jobs only for hired candidates. Larger datasets in addition to improving performance general features of a model also allow for a more robust assessment of fairness.

The results obtained clearly show the need to balance performance and *fairness.* Models that show very high performance, when tested on discriminant data, are not necessarily the best, the accuracy in fact highlights their ability to emulate the human discrimination.

Conversely, a slight reduction in performance does not necessarily lead to a less effective model, but could indicate that the algorithm is selecting candidates solely based on their skills, without being influenced by human biases present in the test set. Fair models may appear to perform less well than models biased, since they generate forecasts that are different from those expected, challenging the dynamics pre-existing discrimination.

However, the real breakthrough is not to mitigate the disparities in the models, which inevitably result less performing if evaluated on *biased test sets,* but give a “good example”, generating datasets more balanced and representative. AI models are not only able to speed up the decision-making processes, but can help make them more equitable and inclusive.

Their strong influence on human thought should not be used to perpetuate inequalities of the past, but to ensure a fairer future for all.

Bibliography

• Aasheim, Tor H, and Knut T Hufthammer. «Bias Mitigation with AIF360: A Comparative Study», sd

• Adecco Group. «The Adecco Group completes acquisition of stake in majority stake in AKKA Technologies». *The Adecco Group* (blog), February 24, 2022. https://adeccogroup.it/the-adecco-group-complete-lacquisition-of-quota-di-majority-of-akka-technologies/.

• Alikhademi, Kiana, Brianna Richardson, Emma Drobina, and Juan E. Gilbert. «Can Explainable AI Explain Unfair no? A Framework for Evaluating Explainable TO THE". arXiv, 14 June 2021. https://doi.org/10.48550/arXiv.2106.07483.

• Bellamy, Rachel KE, Kuntal Dey, Michael Hind, Samuel C. Hoffman, Stephanie Houde, Kalapriya Kannan, Pranay Lohia, et al. «AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias". arXiv, 3 October 2018. https://doi.org/10.48550/arXiv.1810.01943.

• Bird, Sarah, Miroslav Dudík, Richard Edgar, Brandon Horn, Roman Lutz, Vanessa Milan, Mehrnoosh Sameki, Hanna Wallach, Kathleen Walker, and Allovus Design. «Fairlearn: A Toolkit for Assessing and Improving Fairness in AI», n.d

• Diversity, Equity & Inclusion. «Akkodis.com», sd

• Fisher Phillips. «New York Lawmakers Aim to Close Loopholes in NYC's AI Bias Audit Law and Add Teeth to Workplace Protections". Accessed 7 March 2025. https://www.fisherphillips.com/en/news-insights/new-york-lawmakers-aim-to-close-loopholes-nycs-ai-bias-audit-law.html.

• Gilliland, Stephen W. «The Perceived Fairness of Selection Systems: An Organizational Justice Perspective". *The Academy of Management Review* 18, issue. 4 (1993): 694–734. https://doi.org/10.2307/258595.

• github. «Countries.csv». Github. Consulted February 24 2025. https://raw.githubusercontent.com/google/dspl/master/samples/google/canonical/countries.csv.

• Goethals, Sofie, David Martens, and Toon Calders. «PreCoF: Counterfactual Explanations for Fairness". *Machine Learning* 113, fasci. 5 (May 2024): 3111–42. https://doi.org/10.1007/s10994-023-06319-8.

• HenryChinaski, Matteo. «Italian-Municipalities-2018-Sql-Json-excel». Github. github. Accessed February 24, 2025. https://github.com/MatteoHenryChinaski/Comuni-Italiani-2018-Sql-Json-excel/tree/master.

• Holzinger, Andreas, Randy Goebel, Ruth Fong, Taesup Moon, Klaus-Robert Müller, and Wojciech Samek, on c. Of. *xxAI - Beyond Explainable AI: International Workshop, Held in Conjunction with ICML 2020, July 18, 2020, Vienna, Austria, Revised and Extended Papers.* Vol. 13200. Lecture Notes in Computer Science. Cham: Springer International Publishing, 2022. https://doi.org/10.1007/978-3-031-04083-2.

• Madaio, Michael, Lisa Egede, Hariharan Subramonyam, Jennifer Wortman Vaughan, and Hanna Wallach. «Assessing the Fairness of AI Systems: AI Practitioners' Processes, Challenges, and Needs for Support". arXiv, February 10, 2022. https://doi.org/10.48550/arXiv.2112.05675.

• Mehrabi, Ninareh, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan. «A Survey on Bias and Fairness in Machine Learning». arXiv, January 25 2022. https://doi.org/10.48550/arXiv.1908.09635.

• Mujtaba, Dena F., and Nihar R. Mahapatra. «Fairness in AI-Driven Recruitment: Challenges, Metrics, Methods, and Future Directions". arXiv, June 2, 2024. https://doi.org/10.48550/arXiv.2405.19699.

• «Packed with Loopholes: Why the AI Act Fails to Protect Civic Space and the Rule of Law | ECNL», 4 March 2024. https://ecnl.org/news/packed-loopholes-why-ai-act-fails-protect-civic-space-and-rule-law.

• Panagiotou, Emmanouil, Arjun Roy, and Eirini Ntoutsi. «Synthetic Tabular Data Generation for Class Imbalance and Fairness: A Comparative Study». arXiv, 8 September 2024. https://doi.org/10.48550/arXiv.2409.05215.

• Patalay, Prathamesh. «COMPAS : Unfair Algorithm?» *Medium* (blog), November 22 2023. https://medium.com/@lamdaa/compas-unfair-algorithm-812702ed6a6a.