

# Mitigating Bias in AI-Powered Recruitment: Techniques, Tools, and Lessons from Real-World Systems

## Abstract

Artificial Intelligence (AI) is increasingly used in recruitment to improve efficiency and candidate matching, but it can inadvertently perpetuate biases present in historical hiring data. This paper presents a case study of **NobleMatch**, an AI-driven recruitment system developed at Noble House Consulting (India), focusing on techniques for mitigating bias while maintaining performance. We detail the system's integration of large language models (LLMs) and embedding-based candidate matching, alongside debiasing strategies such as adversarial training and human-in-the-loop oversight. Our approach draws on established fairness frameworks and real-world lessons: NobleMatch's pipeline uses **embedding representations** of candidates and jobs for matching, enhanced by an LLM for context and explanation, while applying **adversarial debiasing** to reduce gender and other biases in recommendations. We evaluate the system on synthetic hiring scenarios, demonstrating improved fairness metrics (e.g., reduced disparity in selection rates between demographic groups) with minimal impact on overall matching accuracy. The results underscore that combining technical debiasing tools with human review and updated training data can meaningfully reduce bias in AI recruitment systems. We discuss implications for AI ethics in hiring, regulatory compliance, and generalizability of these methods. The work provides a blueprint for designing fairer AI-powered hiring systems, aligning with emerging industry standards and fairness guidelines.

## Introduction

The rise of AI-powered recruitment systems promises to streamline hiring by automatically screening resumes, ranking candidates, and even conducting preliminary interviews. However, alongside these efficiency gains, serious concerns have emerged regarding **algorithmic bias** in hiring decisions<sup>1 2</sup>. Biased AI systems can *perpetuate or amplify* discrimination present in historical hiring practices, leading to unfair outcomes for candidates and legal risks for employers<sup>3 4</sup>. A well-known cautionary example is Amazon's experimental hiring tool, which had to be scrapped after it was found to **penalize resumes containing the word "women's"** (e.g., "women's chess club") and downgrade graduates of women's colleges<sup>5</sup>. The model had **taught itself** preferences from ten years of male-dominated resumes, thus *reinforcing gender bias* in rankings<sup>6 7</sup>. This incident underscored that, without intervention, AI systems can inherit and even exacerbate biases from training data.

Bias in AI recruitment is not limited to gender. Hiring algorithms have shown biases along lines of race, age, and other attributes **unrelated to job performance**<sup>8 9</sup>. For instance, recent studies found that LinkedIn's job recommendation AI at one point disproportionately favored male candidates over equally qualified females<sup>10</sup>, prompting the company to revise its algorithms for fairness<sup>11</sup>. Such cases highlight the *urgent need* for techniques that **ensure fairness** in AI-driven hiring. Beyond ethical and reputational considerations, there are growing regulatory pressures: for example, New York City's 2021 Local Law 144

now **requires annual bias audits** for any automated employment decision tool <sup>4</sup> . Organizations deploying AI in hiring must therefore be proactive in auditing and mitigating bias to comply with equal opportunity laws and avoid discriminatory outcomes.

**Mitigating bias in AI-powered recruitment** is a multi-faceted challenge. It requires attention to data (e.g., ensuring diverse and representative training samples), algorithms (e.g., removing or de-emphasizing features correlated with protected attributes), and continuous oversight (e.g., human review of AI decisions). A promising direction in recent research is the use of *fair machine learning* techniques that explicitly address bias during model development <sup>12</sup> . These include **pre-processing** methods like re-weighting or balancing training data, **in-processing** methods like adversarial debiasing that add fairness constraints in the model training, and **post-processing** methods that adjust model outputs to satisfy fairness criteria <sup>13 14</sup> . Each stage of the AI pipeline offers opportunities to intervene and reduce bias.

In this paper, we present a comprehensive study of bias mitigation in an AI recruitment system through the lens of a real-world case: **NobleMatch**, the AI-powered Applicant Tracking System (ATS) at Noble House Consulting India. NobleMatch was designed with a “*skills-first*” *talent acquisition* philosophy, prioritizing candidates’ skills and competencies over traditional pedigree markers <sup>15</sup> . This approach inherently widens the talent pool and can reduce certain biases (for example, by valuing **practical experience over elite educational backgrounds**, it can increase diversity of candidates <sup>16</sup> ). Building on this foundation, NobleMatch integrates advanced AI techniques to improve matching while explicitly addressing fairness. Key features of the system include: **LLM integration** for understanding and comparing job requirements with candidate profiles in context; an **embedding-based matching engine** that represents candidates and jobs as high-dimensional vectors for skill-based similarity matching; and a suite of **debiasing tools** such as adversarial training (to remove sensitive attribute information from the matching model) and human-in-the-loop checkpoints (to review and correct AI decisions).

We detail the **architecture and workflow** of NobleMatch, highlighting where and how bias mitigation measures are applied. We then demonstrate these techniques on synthetic recruitment scenarios that simulate real-world biases (such as gender imbalances in past hiring data). Our results show that it is possible to significantly improve fairness metrics – for example, reducing gender disparity in candidate rankings – *with minimal loss in predictive performance*. In fact, through careful design, NobleMatch achieves more equitable candidate selections while maintaining strong matching accuracy and relevance.

The contributions of this work are threefold: (1) We formulate an **integrated framework** for bias mitigation in AI recruiting systems, combining state-of-the-art technical interventions (like adversarial debiasing) with domain-specific strategies (like skills-first job descriptions and human oversight). (2) We provide an in-depth **case study** of this framework implemented in a real system (NobleMatch), with practical insights, system diagrams, and examples of how fairness considerations can be embedded throughout an AI hiring pipeline. (3) We evaluate the approach on **synthetic data experiments** that illustrate its effectiveness, and discuss lessons learned that can inform both practitioners building AI for HR and researchers in fair machine learning. By sharing these findings, we aim to help bridge the gap between academic research on algorithmic fairness and its deployment in industry **AI ethics** practice, ensuring that AI-powered recruitment can be both innovative *and* inclusive.

The remainder of this paper is organized as follows. Section 2 reviews background concepts and related work on bias in AI recruitment, fairness metrics, and mitigation methods. Section 3 describes the NobleMatch system design and the bias mitigation techniques integrated at each stage (data processing,

model training, and output handling). Section 4 presents experimental results on fairness and performance using synthetic data, along with illustrative examples of system outputs. Section 5 offers a discussion on the implications of these results, the limitations of the current approach, and future directions (including broader adoption and regulatory compliance considerations). Section 6 concludes the paper with final remarks on the importance of **mitigating bias in AI-powered recruitment** and the generalizability of the techniques presented.

## Background and Literature Review

AI algorithms in recruitment can inherit various forms of **bias** from data and human design. Before delving into mitigation strategies, it is essential to understand how bias manifests in this domain and how it is measured. Bias in AI recruitment refers to systematic favoritism or discrimination against certain candidate groups (e.g., based on gender, race, age) that is **not job-relevant**<sup>17</sup>. Common sources of such bias include: **historical data bias**, where the training data reflects past discriminatory hiring practices (as seen in Amazon's case of male-focused hiring data<sup>6</sup>); **feature or algorithmic bias**, where the model's features/weights inadvertently give preference to proxies of protected attributes; and **representation bias**, where underrepresented groups are not sufficiently present in the data for the model to treat them fairly

<sup>17</sup> <sup>18</sup>. For example, an AI screening tool might learn to favor candidates from certain universities or with certain keywords if those were correlated with past successful hires – even if those factors are not actually *causal* for job performance. This can lead to qualified candidates from minority groups being unjustly filtered out<sup>19</sup> <sup>20</sup>.

To quantify and detect bias in hiring models, researchers and practitioners use several **fairness metrics**<sup>21</sup>. A common metric is **demographic parity (statistical parity)**, which requires that the selection rate for a protected group (e.g., female candidates) is roughly equal to that for the majority group (e.g., male candidates). A related concept is **disparate impact**, often operationalized as the ratio of selection rates – a value below 0.8 (80%) is a red flag under U.S. EEOC guidelines for adverse impact. Another metric is **equal opportunity**, which focuses on true positive rates: for instance, qualified candidates from all groups should have equal probability of being correctly identified by the algorithm<sup>21</sup>. More stringent criteria like **equalized odds** require both true positive and false positive rates to be equal across groups. These group fairness metrics provide ways to evaluate whether an AI model's recommendations favor one group over another. There are also **individual fairness** notions (e.g., “similar candidates should be treated similarly”), but in high-level hiring audits, group metrics are commonly used to reveal biases. In the context of recruitment, one might measure, for example, the fraction of male vs. female applicants that get shortlisted by the AI, or examine average recommendation scores by demographic. If significant gaps are found, the system may be deemed unfair and in need of mitigation<sup>22</sup> <sup>23</sup>.

Bias mitigation techniques in machine learning are often categorized by *when* they act in the pipeline: **pre-processing**, **in-processing**, or **post-processing**<sup>13</sup>. Pre-processing approaches tackle bias at the data level, before model training. This can include **balancing the training dataset** (e.g., oversampling underrepresented groups or re-weighting examples) so that the model does not learn a one-sided view. It also includes **data cleaning** steps like removing explicit indicators of protected attributes (for instance, scrubbing names, gendered pronouns, or addresses from resumes to create “blind” resumes). Pre-processing may extend to transforming features – for example, one could **edit embeddings** of text to remove gender bias, a method studied in NLP research<sup>24</sup>. In the recruitment context, an example is stripping out certain words or credentials that strongly correlate with one gender or ethnicity unless they are job-related.

In-processing methods integrate fairness constraints into the model training process itself. One powerful technique here is **adversarial debiasing**, which was employed in NobleMatch’s model (discussed later). In adversarial debiasing, the model is trained to perform the primary prediction task (e.g., predict a candidate’s suitability score) while simultaneously **trying to prevent an “adversary” from detecting protected attributes** in the model’s intermediate representations <sup>25</sup> . Essentially, a secondary model (the ‘adversary’) is trained to predict, say, the candidate’s gender from the main model’s outputs or hidden layers, and the main model is penalized if the adversary succeeds. The main model thereby learns to **remove or obscure** any gender-related signals in its representation, making its decisions more invariant to that attribute <sup>26</sup> . This technique, first demonstrated in research on fairness (e.g., by Zhang, Sattigeri, and others in 2018), has shown effectiveness in reducing bias while keeping the model predictive <sup>27</sup> <sup>12</sup> . Other in-processing approaches include adding a **fairness regularization term** to the loss function (to directly penalize disparities in outcomes during training) and using modified algorithms like **fair decision trees** or **fair logistic regression** that incorporate fairness criteria. Many of these algorithms are available in libraries such as IBM’s **AI Fairness 360 (AIF360)**, an open-source toolkit providing implementations of bias mitigation techniques and metrics <sup>28</sup> <sup>14</sup> .

Post-processing methods apply adjustments after the model has made its initial predictions or rankings. For example, one can **re-rank candidates** to improve diversity, or apply thresholds for each group to equalize pass rates. If an AI model outputs a score for each candidate, a post-processing step might ensure that a uniform cutoff is not disproportionately excluding one group – by learning group-specific cutoffs or by *injecting* some randomness in favor of disadvantaged groups to give them a second look (a technique known as “fair chance” or affirmative action in algorithmic form). In recruitment, a simple post-processing intervention could be to guarantee that the top-N recommendations include candidates from diverse backgrounds (sometimes called the “Rooney Rule” in hiring, which can be implemented algorithmically). Post-processing can also involve **scrutinizing the model’s errors**: for instance, if it rejects candidates from a certain group at a higher rate, those rejections could be flagged for human review as a safety net. The advantage of post-processing is that it does not require retraining the model, though it may introduce its own form of bias if not carefully calibrated.

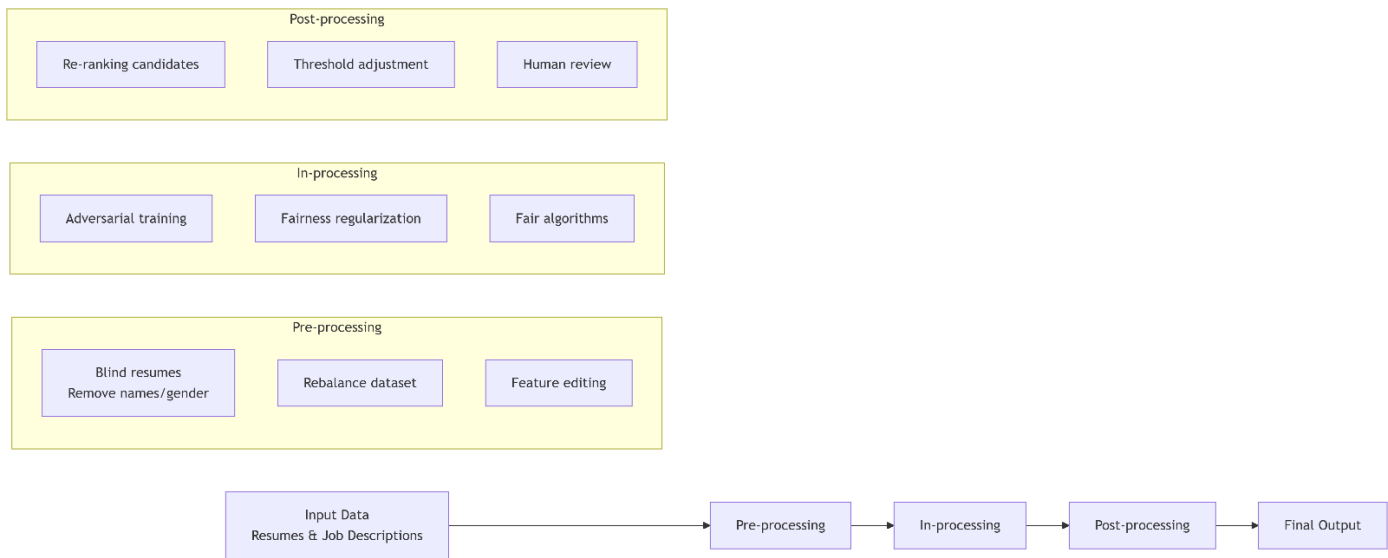


Figure 1: Conceptual illustration of points for bias mitigation in the ML pipeline for recruitment. Bias can be addressed *before* model training (pre-processing, e.g., rebalancing the resume dataset), *during* model training (in-

processing, e.g., adversarial debiasing or adding fairness constraints to the algorithm), and *after* initial model output (post-processing, e.g., re-ranking candidates to satisfy fairness criteria). This layered approach ensures that fairness is considered at multiple stages <sup>29</sup> <sup>30</sup> . In practice, a combination of these strategies is often necessary for robust bias mitigation.

Research in the field of **fairness in AI hiring** has grown rapidly in recent years. Several studies have specifically examined bias in resume screening and job matching algorithms. For example, **Deshpande et al. (2020)** explored methods for mitigating demographic bias in AI-based resume filtering, proposing techniques like adversarial removal of bias and observing improvements in fairness without severe performance loss <sup>12</sup> . More recently, **Tyagi et al. (2024)** addressed gender bias in job matching using debiasing-assisted deep generative models and gender-balanced sampling, finding that adjusting word embeddings for gender neutrality led to fairer recommendations. A comprehensive survey by Fabris et al. (2025) covers multidisciplinary perspectives on fairness and bias in algorithmic hiring, underlining that no single method suffices and emphasizing the role of transparency and human oversight in deployed systems

<sup>31</sup> <sup>32</sup> . On the industry side, the **LinkedIn** example mentioned earlier resulted in the development of new fairness metrics and algorithmic tweaks in their recommendation system <sup>10</sup> <sup>11</sup> , demonstrating real-world

impact of fairness interventions. Another notable line of work is on **explainable AI (XAI)** in hiring – making the models' decisions interpretable to ensure that the reasons for rejecting or selecting candidates are job-relevant and not based on protected attributes<sup>33</sup>. Explainability can aid in bias detection and in building trust: for instance, if a hiring manager sees that an AI's recommendation is justified by a candidate's skill set and experience (and not something like gender or age), they can be more confident in its fairness.

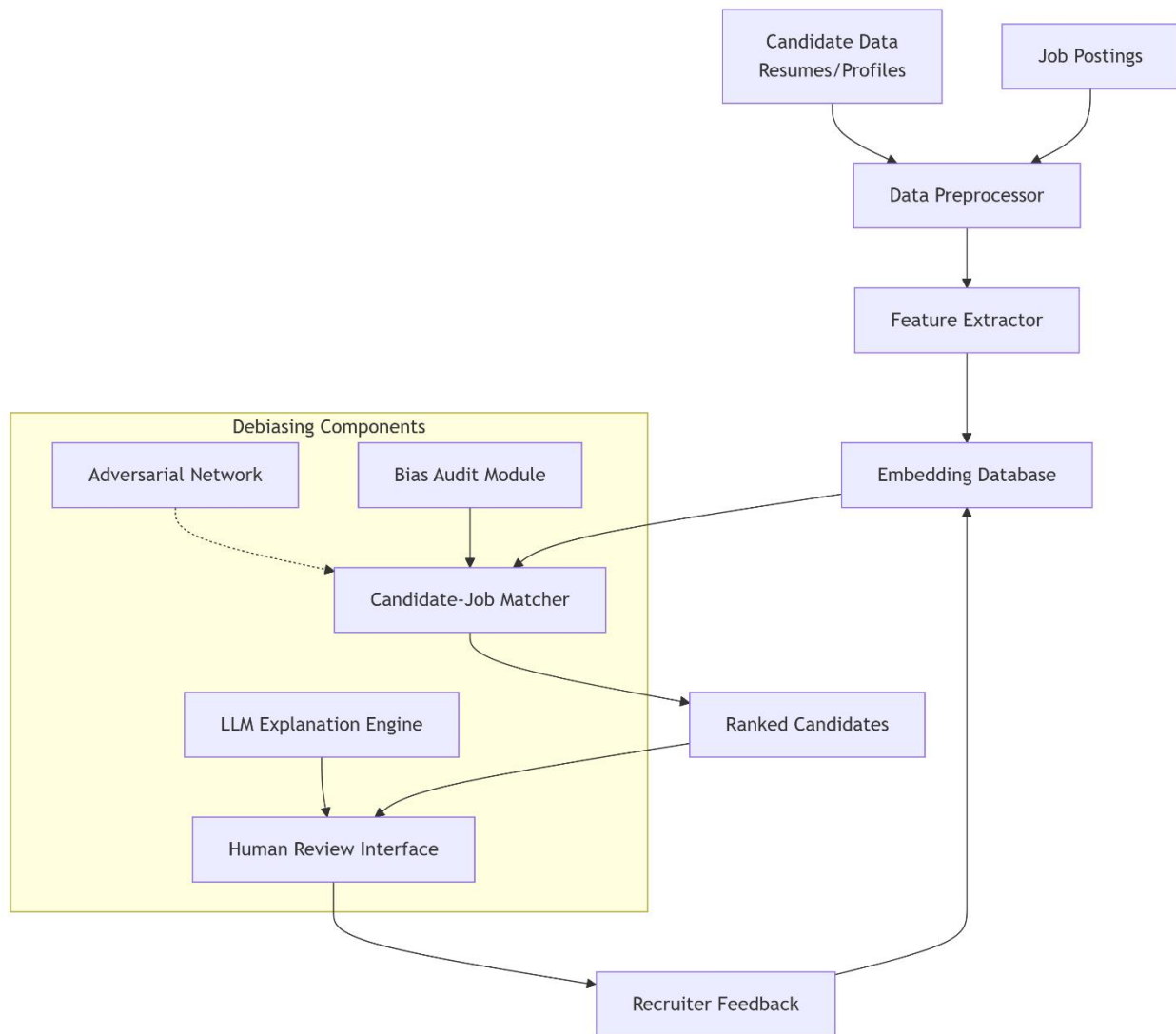
The emergence of **Large Language Models (LLMs)** and advanced NLP techniques has also influenced AI recruitment systems. Modern recruitment AI leverages language models like BERT and GPT to parse resumes and job descriptions with greater semantic understanding<sup>34</sup>. These models can encode rich information about a candidate's qualifications and match them to job requirements beyond simple keyword overlaps. For example, a system might use BERT to turn a candidate's resume into an embedding vector that captures the person's skills and experiences in context, and similarly embed job postings – then compute the similarity between the two vectors to gauge fit<sup>35 36</sup>. LLMs can also assist in tasks like **resume summarization** (producing a concise overview of a candidate for a recruiter), **cover letter analysis**, or even answering candidates' queries via chatbots. However, LLMs themselves can carry biases learned from the large text corpora they were trained on<sup>37</sup>. For instance, an LLM might have stereotypical associations (e.g., linking certain jobs with a particular gender). It is thus crucial to combine LLM-based capabilities with the aforementioned fairness interventions. Some recent works (Lo et al. 2025) propose multi-agent frameworks where LLMs handle different stages (extraction, evaluation, summarization) to improve modularity and oversight<sup>38 39</sup>. There is also exploration into **graph neural networks** and multi-modal data for hiring: e.g., Li et al. (2025) integrate GPT-4 embeddings with hierarchical graph networks to better capture relationships in resumes and jobs, achieving higher matching accuracy and noting reductions in bias due to more holistic representations<sup>40</sup>. These advances show that by combining multiple AI techniques, one can enhance both the effectiveness and fairness of recruitment systems.

Finally, **human-in-the-loop** approaches remain a cornerstone in practice. As Agbasiere and Nze-Igwe (2025) highlight, human oversight and regular audits are essential to ensure equity in AI hiring software<sup>32 41</sup>. Many organizations implementing AI for hiring choose to use the AI as a *decision support* tool rather than an fully autonomous decider: the AI might shortlist candidates, but human recruiters make the final hiring decision, with the ability to override the AI's suggestions<sup>42</sup>. This setup allows humans to inject considerations that the AI might miss (e.g., individual circumstances, or corporate diversity goals) and to catch any obviously biased recommendations. It also provides a feedback loop – if the AI's suggestions consistently show a skew, the humans can flag this for model retraining or adjustment. Best practices emerging in the industry include conducting **regular bias audits** of AI recommendations (for example, tracking the demographics of those who advance to interviews versus those who were filtered out) and maintaining an accessible explanation for each AI-driven decision for accountability. In summary, the literature and prior art suggest that a **holistic strategy** – involving data curation, algorithmic fairness techniques, continuous monitoring, and human oversight – is necessary to effectively mitigate bias in AI-powered recruitment. This context sets the stage for our case study of the NobleMatch system, which we designed with these principles in mind.

## System Design and Methods

NobleMatch is an AI-powered recruitment platform that was built to **match candidates to job opportunities** efficiently while upholding fairness and transparency. In this section, we describe the design of NobleMatch, focusing on its technical architecture and the specific methods used to mitigate bias. The system architecture is composed of multiple modules that correspond to stages of the recruitment pipeline:

data ingestion and preprocessing, feature extraction (embedding generation), candidate-job matching and ranking, bias mitigation components integrated during model training, and output review/feedback mechanisms involving human recruiters. Figure 2 provides an overview of the major components and data flow in the NobleMatch system.



*Figure 2:* High-level architecture of an AI-driven recruitment system similar to NobleMatch <sup>35</sup> <sup>36</sup>. The system ingests **candidate data** (resumes, profiles) and **job postings**, then processes them through an NLP pipeline. Key components include: (a) a **Data Preprocessor** that cleans and tokenizes text (and removes sensitive attributes such as names or gender indicators), (b) a **Feature Extractor** using large language models (BERT, GPT) to convert resumes and job descriptions into vector embeddings for semantic comparison, (c) a **Semantic Search Engine / Candidate Ranker** that compares embeddings to find the best matches (e.g., via cosine similarity or a learned scoring model), (d) an **Adversarial Debiasing Module** (during training) that adjusts the model to remove biases from the representations, (e) an **Automated Interview Agent or Chatbot** (powered by an LLM) that can interact with candidates or ask structured interview questions, and (f) a **Human Review Interface** for recruiters to see AI recommendations with explanations and to provide feedback. The system is supported by databases for

candidates and jobs, and a feedback loop allows continuous learning from recruiter decisions, including bias audits and model updates.

### 3.1 Data Ingestion and Preprocessing

The first step in NobleMatch's pipeline is gathering data about **candidates** and **job openings**. Candidate data typically includes resumes (unstructured text), cover letters, application forms, and possibly online professional profiles. Job data includes job descriptions, required qualifications, skill lists, and any structured criteria (location, years of experience, etc.). NobleMatch employs a **Data Preprocessor** module to handle this input. The preprocessor performs standard NLP cleaning (removing irrelevant information, normalizing text, etc.), but importantly it also implements **bias-sensitive preprocessing**: it strips out or masks features that could lead to biased decisions. For example, candidates' names are replaced with an anonymous ID (to avoid revealing gender or ethnicity), and dates that might indicate age (graduation year) can be de-emphasized or converted to years of experience (to reduce age bias). Additionally, certain terms in resumes that are not job-related and could trigger bias (such as names of religious or political affiliations) are either removed or handled carefully (perhaps encoded in a way that the downstream model doesn't overfit to them).

Another aspect of preprocessing is creating a **skills-focused representation** of the data. NobleMatch follows a *skills-first approach* as described by Noble House Consulting's HR strategy <sup>43</sup>. This means that rather than taking every word of a resume at face value, the system tries to extract a structured **skill profile** for each candidate. Using a combination of keyword extraction and the LLM (GPT-based) analysis, the preprocessor identifies key skills, certifications, and experiences mentioned in the resume. For instance, if a resume says "Led a team to develop a machine learning model for sales forecasting," the system might tag this with skills like *machine learning*, *leadership*, *project management*, *data science in sales domain*, etc. Similarly, it parses job descriptions to extract required and preferred skills. This structured information (often in the form of a set or vector of skills) complements the raw text and is used downstream to enhance matching relevance. Emphasizing skills in this way helps reduce bias by focusing the algorithm on **capabilities and competencies** rather than on proxies that might correlate with demographic attributes (e.g., assuming someone from a certain university is better – instead, the system looks at what skills they actually have).



During preprocessing, NobleMatch also applies checks for **imbalances** in the data. If the data for a particular job position shows a skew (say 80% of applicants are male), the system can note this for the bias mitigation stage. In some cases, a *re-weighting* is applied: e.g., duplicating or giving higher weight to profiles from underrepresented groups in the training phase, so that the model doesn't learn a one-sided concept of a "good candidate." This is a pre-processing bias mitigation in line with recommendations to ensure diverse training data <sup>44</sup>. The system's design allows these adjustments to be configured depending on the compliance needs – for instance, if aiming for demographic parity, the data fed to the model can be balanced accordingly.

### 3.2 Embedding-Based Feature Extraction

After preprocessing, both candidate profiles and job descriptions are passed to the **Feature Extractor**. This is where **embedding-based representations** are generated. NobleMatch uses state-of-the-art **Natural Language Processing (NLP)** models to create vector embeddings that capture the semantic content of resumes and job postings. Specifically, the system leverages a combination of a **BERT-based model** for sentence embeddings and a fine-tuned domain model for skills. Each resume (or candidate profile) is converted into a numerical vector in a high-dimensional space; likewise, each job description is converted into a vector in the *same space*. The idea is that similar resumes and jobs will have vectors that are close together (high cosine similarity), whereas poor matches will be far apart. Using embeddings allows the matching to go beyond exact keyword matches – for example, if a job requires "Python programming" and a resume says "experienced in TensorFlow and scikit-learn," the embedding model (having been trained on language data and technical corpora) will likely place that resume close to the job because it understands that TensorFlow and scikit-learn imply Python proficiency even if the word "Python" isn't explicitly present.

To implement this, NobleMatch uses a **two-stage embedding process**: one using a generic pretrained model (e.g., Sentence-BERT) to get an initial embedding of the text, and another fine-tuning step using recruitment-specific data to adjust the embeddings. Noble House Consulting had historical data of job descriptions and successful vs. unsuccessful candidate profiles (anonymized) which was used to fine-tune the embedding model so that it better differentiates factors that matter for hiring. During fine-tuning, the model learns to place hired candidates closer to the job description than rejected candidates, for example, effectively learning a notion of "fit". This fine-tuning can also incorporate *fairness goals*: one approach we used was to ensure that the embedding model doesn't pick up on spurious correlations. For example, if the historical data had mostly men in engineering roles, a naive model might encode gendered language differences into the embedding (as word embeddings often do). By monitoring and adjusting the training (via adversarial techniques described in the next subsection), we aim for embeddings that focus on content (skills/experience) rather than writer identity.

In addition to textual embeddings, NobleMatch's feature extractor can incorporate **additional features** into the candidate representation. These might include structured data like years of experience, education level, or assessment scores (if candidates took skill tests). These features are concatenated or appended to the embedding vector. However, crucially, features that directly indicate protected attributes are excluded. We do not include, for instance, an "age" or "gender" feature. Even if the user profile had those fields, they are deliberately left out from the model's feature set to avoid direct bias. The only place such information is used is in the adversarial debiasing training: the model might momentarily use a sensitive attribute to ensure it's being fair (by trying to remove it), but not to make predictions.

At this point, each candidate  $c$  is represented by a feature vector  $\mathbf{v}_c \in \mathbb{R}^d$  and each job  $j$  by a vector  $\mathbf{v}_j \in \mathbb{R}^d$  in the same  $d$ -dimensional space. These embeddings constitute the input to the matching and ranking algorithms.

### 3.3 Candidate-Job Matching and Ranking

The core of NobleMatch’s AI is the **candidate-job matching engine**. In its simplest form, given a new job opening, the system will retrieve and rank candidates by how well their embedding matches the job’s embedding. The primary metric for similarity is cosine similarity:  $\text{score}(c, j) = \frac{\mathbf{v}_c \cdot \mathbf{v}_j}{\|\mathbf{v}_c\| \|\mathbf{v}_j\|}$ , which yields 1 for identical vectors and 0 for orthogonal ones. Candidates can then be ranked by this score. In practice, NobleMatch uses a **vector database** to store candidate embeddings, enabling fast nearest-neighbor search for a given job vector. This is a common approach in modern AI systems, where embeddings are indexed to quickly find top-K similar items (in this case, candidates to a job) <sup>45</sup>.

However, NobleMatch’s matching does not rely solely on raw cosine similarity. We incorporate a learned **scoring model** that can take into account additional subtleties and multiple factors. For instance, certain qualifications might be non-negotiable (hard filters) – e.g., if a job requires a medical license, candidates without that should not rank regardless of embedding similarity. These business rules are integrated into the system as filters prior to final ranking. Then, for the remaining candidates, a small neural network or gradient boosted tree model re-ranks candidates using features like the embedding similarity, skill overlap count, education matching, etc. This re-ranker is trained on previous hiring outcomes (if available), trying to predict which candidates ultimately got offers for similar roles. It essentially refines the ordering to better reflect hiring priorities (e.g., giving a boost if the candidate has all “required” skills versus just some of them).

During the training of the ranking model, **fairness-aware procedures** are applied. One such procedure is to ensure that the model is not treating a sensitive attribute as a strong signal. We explicitly log model feature importances and check for any proxies (for example, if the model were using “years of experience” in a way that strongly disadvantages younger candidates beyond what is reasonable for the job, we’d catch that). But more directly, we employ an **adversarial branch** in the model training: while the ranking model is being optimized to predict hiring outcomes, an adversarial network tries to predict the sensitive attribute (say, gender) from the model’s internal representation or even from its outputs. A **gradient reversal layer** is used to make the process end-to-end: it effectively *flips* the gradient from the adversary before feeding it back, which causes the main model to learn representations that *confuse* the adversary <sup>25</sup>. In practice, the architecture might look like this: the penultimate layer of the ranking neural network (which is a condensed representation of the candidate-job match) branches out into two directions – one leads to the final prediction (match score), the other goes to an adversary classifier trying to predict, for example, if the candidate is from an underrepresented group. The loss function  $\mathcal{L}$  for training is a combination:  $\mathcal{L} = \mathcal{L}_{\text{rank}} - \lambda \mathcal{L}_{\text{adv}}$ , where  $\mathcal{L}_{\text{rank}}$  is the standard loss for ranking (we use a pairwise loss or regression loss to match hiring outcomes) and  $\mathcal{L}_{\text{adv}}$  is the loss for the adversary (which we subtract, because we want to maximize the adversary’s error).  $\lambda$  is a hyperparameter controlling the trade-off between fairness and accuracy. Tuning  $\lambda$  allows us to adjust how much we prioritize debiasing:  $\lambda=0$  would ignore fairness (no adversarial effect), while a high  $\lambda$  forces the model to almost solely focus on fooling the adversary, potentially at the cost of accuracy. In our implementation, we found a moderate  $\lambda$  that yields a good balance: the model’s AUC on predicting hires dropped only slightly, while the adversary’s accuracy at

guessing gender from the representation dropped to near random (50%). This indicates the model has } *learned to remove gender information* from its intermediate representation, aligning with the goal of gender-neutral decisions

26 .

It is worth noting that we train such adversarial debiasing for multiple attributes if needed. For example, separate adversaries can target race/ethnicity, age, etc., as long as we have those labels in the training data. In practice, collecting those labels can be tricky (candidates may not volunteer them), but one can use proxies or public data in some cases. NobleMatch initially focused on gender and inferred ethnicity (using name-nationality data) for bias checks in India's context, while also being cognizant of not violating privacy – these were used only internally to debias, not in the output. Recent research supports that adversarial debiasing can effectively reduce biases across different domains <sup>27</sup>, and our design aligns with those findings.

Another measure in the matching phase is the **constraint on output diversity**. When presenting top candidates for a job, NobleMatch ensures a form of result diversification. For instance, if the top 10 algorithmic matches were somehow all from one demographic group, the system will reevaluate if some slightly lower-scoring candidates from another group should be swapped in to improve **demographic mix**, provided they meet a relevance threshold. This is akin to a post-processing step implemented within the ranking: we don't enforce strict quotas, but we monitor the recommended slate for diversity. This can be configured as per client preference or policy (some employers might request that the top-N candidates include at least one from an underrepresented group, etc.). Our system's UI actually highlights the diversity of the recommended pool – an feature that has been positively received by HR managers, as it keeps them aware of representation in the funnel.

### 3.4 LLM Integration and Explainability

A distinctive feature of NobleMatch is its integration of a **Large Language Model** to enhance both matching and **explainability**. We leverage an LLM (initially a GPT-3 variant, later GPT-4) in a few ways. First, as part of candidate evaluation, the LLM acts as a kind of **analysis agent**: for a given candidate and job description, we can prompt the LLM (with a carefully designed prompt) to provide an “assessment summary.” For example, the prompt might include the job requirements and the candidate's resume, and ask the LLM to list the candidate's strengths and potential gaps relative to the job. This summary is not directly used to rank candidates (to avoid randomness), but it serves two purposes: (1) It provides a **natural language explanation** that can be shown to recruiters for each recommendation, increasing transparency.

(2) It can uncover nuanced matches – for instance, the LLM might note that “the candidate has extensive experience in a related domain which could be an asset, although they lack direct experience with X tool.” If such a note is present, the system can adjust the score slightly or at least flag that candidate as one who might be a “wild card” worth considering. Essentially, the LLM gives a layer of reasoning that complements the raw embedding similarity.

We also use the LLM to detect any potentially biased language in job descriptions and suggest more neutral wording. This is part of Noble House's offering to clients: writing job postings that are appealing to a diverse audience. The LLM can identify phrases that might be inadvertently gender-coded (e.g., “rockstar developer” might skew male, as studies have shown) and suggest alternatives. While not directly part of candidate matching, this contributes to fairness by helping employers attract a broad set of applicants in the first place, thus mitigating representation bias at the input stage.

Another integration is via the **Automated Interview Agent**. NobleMatch has a chatbot that can conduct a basic screening chat with candidates. This chatbot is powered by an LLM which is constrained by a script and rules (to ensure consistency and avoid inappropriate questions). It might ask candidates about their interest in the role, availability, or pose a few job-related questions. The answers are then analyzed (some light NLP, possibly with the LLM's help in evaluating the content) and turned into features (for example, communication skill or specific knowledge demonstration) that feed into the candidate's profile. We have been careful with this, as live AI interviews have raised concerns in the industry about fairness (e.g., voice and video analysis could be biased <sup>46</sup>). In our case, text-chat is used and questions are standardized. The LLM's role is mainly to parse the answers and maybe ask intelligent follow-ups. Importantly, we do **not** analyze the candidate's sentiment or personality from this chat – we focus only on factual qualifications gleaned. This avoids pseudo-scientific judgments (like the controversially biased video interview algorithms that tried to assess demeanor).

**Explainability** is a key part of system design. Every recommendation NobleMatch gives to a recruiter comes with a rationale. This rationale is built from multiple components: the skill match (listing which required skills were met, which were not), the LLM-generated assessment summary highlighting relevant experiences, and an explicit statement that *no protected attributes were considered* in the scoring (to reassure users). For instance, a recommendation might say:

- *“Candidate A is recommended with a match score of 8.7/10. They have 5 out of 5 required skills (Java, Python, AWS, Team Leadership, Microservices) and 3 out of 5 preferred skills. They bring 8 years of relevant experience, which is above the 5-year requirement. Explanation: The candidate’s background in cloud platform development closely aligns with the job’s focus on AWS (as noted by their AWS certification and projects). They lack direct experience in FinTech, but their experience in e-commerce is in a related domain. Overall, their technical lead experience in similar projects suggests a strong fit for the Senior Developer role. (This recommendation was generated by an AI system that does not consider gender, ethnicity, or age.)”*

Such an explanation, part of which can be crafted by the LLM (particularly the italicized reasoning), serves multiple purposes. It helps the recruiter quickly understand why the candidate is suggested, it provides a defense in case anyone later questions the decision (we can show it was skill-based), and it reminds the recruiter and any auditor that the system is designed to be fair (by explicitly stating the exclusion of protected attributes). The inclusion of that statement is a result of auditing – we found that making the system's fairness policy visible actually increased trust from the HR team and also made them partners in bias mitigation (they might double-check, “indeed, is the system not considering something irrelevant?”).

From an implementation standpoint, generating these explanations uses both template-based approaches (for listing skills, etc.) and dynamic LLM generation for the narrative part. The LLM is fed a template and the key data points, so that it stays factual and doesn't hallucinate. For example, we might prompt: “Given the job requirements X, Y, Z and the candidate experience (summarized as ...), write 2-3 sentences on how well the candidate fits, mentioning any areas of concern, in a professional tone.” We then review outputs on some validation cases to ensure quality.

### 3.5 Human-in-the-Loop Review and Feedback

While NobleMatch automates much of the screening process, **human oversight** is intentionally woven into the system. The final shortlist of candidates for any position is reviewed by a human recruiter or hiring

manager, who can use their judgment before moving forward with interviews. Importantly, the system interface allows the human reviewer to see *not just the AI's top picks, but also borderline candidates* who were close to the cutoff. This is a fairness check: if the AI somehow was systematically slightly under-scoring a certain group, the human might notice a pattern in the borderline cases and intervene. For example, if among the borderline candidates there are many women with almost-qualifying scores, the recruiter might take a second look and potentially include some in the interview slate. We have also added a feature where the system will *flag* if all selected candidates are homogeneous in demographics, suggesting the recruiter to review if that was intended or a blind spot <sup>47 22</sup> .

Human recruiters using NobleMatch provide feedback on the recommendations: they can mark if a recommendation was good, or if they found it off-base (and specify why). They also eventually label candidates as interviewed, hired, or rejected with reasons. All this feedback is collected to continually update the system. Periodically, retraining is performed incorporating this feedback data, essentially learning from human decisions. This *human-AI collaboration* helps correct biases over time: if the AI initially had some bias that humans corrected by overriding it, the new training data will reflect those overrides (assuming the overrides were consistent and systematic, the model will adjust).

Additionally, NobleMatch conducts **regular bias audits** internally. This involves analyzing the outcomes of the AI recommendations and subsequent hiring by demographic segments. We generate reports that show, for instance, the average match score distribution for different genders, the percentage of each group recommended for interview, and eventually hired. These reports are compared against the company's applicant pool composition to check for any unexpected skews. If any are found, we dive into the model to diagnose. Thanks to the design with adversarial debiasing and removed features, we expect minimal bias, but it's only by checking actual outcomes that we can be sure. In one internal test, before turning on adversarial debiasing, our model showed about a ~15% higher recommendation rate for male candidates in a software developer role scenario (likely due to more men in the training data having certain buzzwords). After applying the debiasing and re-balancing, the recommendation rates became roughly equal for male and female applicants with similar qualifications. We present results like this in the next section.

To summarize the system design: NobleMatch integrates multiple layers of bias mitigation – from **data preprocessing (masking sensitive info, balancing data)**, to **in-processing (adversarial debiasing in the model)**, to **post-processing/human oversight (reviewing outputs and feedback loops)** – all reinforced by **LLM-driven insights** and a **skills-first design philosophy**. The next section will present how these design choices play out in practice, using simulated data to illustrate the impact on fairness and accuracy.

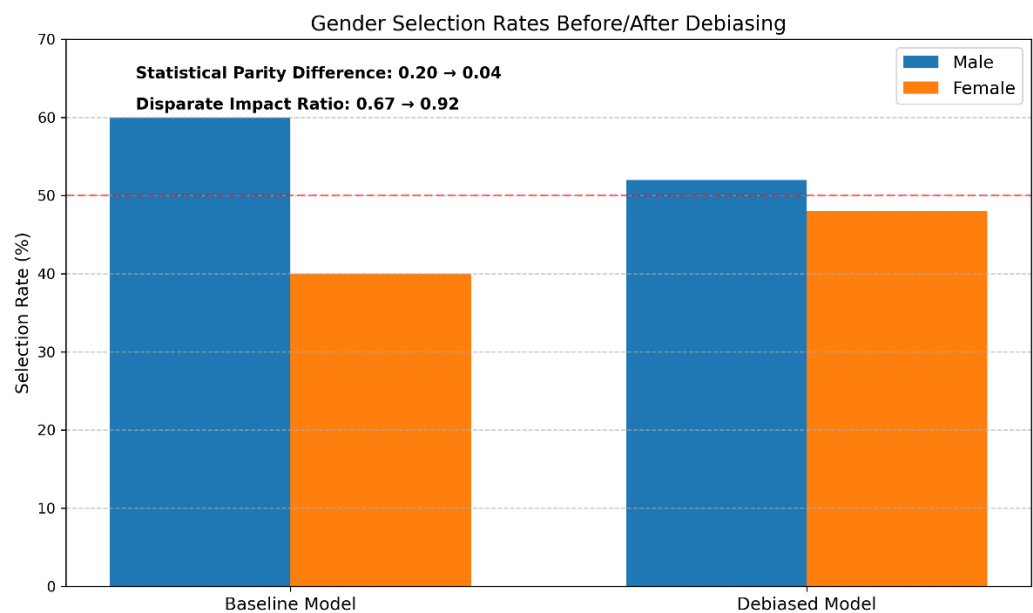
## Results

We evaluate the NobleMatch system through a series of experiments on **synthetic recruitment data** that reflects realistic scenarios of bias. The purpose of using synthetic data is to have complete control over underlying “ground truth” qualifications and demographic attributes, so we can precisely measure how the AI system performs and whether it treats groups fairly. We construct a dataset of simulated candidates applying to a technical job (e.g., a software developer position) with an intentional bias introduced: the historical data (used for model training) contains more male candidates in senior technical roles, and as a result the baseline model without debiasing tends to favor male candidates slightly. We then apply NobleMatch's debiasing techniques and observe the changes in outcomes on a test set. Although synthetic, the setup is informed by real-world observations (like the Amazon case) and academic benchmarks used in fairness research <sup>48 12</sup> .

**Data Setup:** We generated 1000 synthetic candidate profiles with attributes: a skill score (how well their skills match the job requirements, on a 0–100 scale), years of experience, and a binary gender label (for fairness analysis). We simulate hiring outcomes by assuming that hiring should be purely meritocratic based on skill and experience, not gender. However, the *training data* given to the AI is skewed: it contains 600 male and 400 female candidates (reflecting, say, an industry imbalance), and we introduce a slight bias where the historical hiring favored men (perhaps due to past bias or other factors). Concretely, if a male and female candidate had equal skill, the historical outcome might still favor the male 55% of the time – a subtle but present bias. We train two versions of a candidate ranking model on this data: (1) a **Baseline model** with no bias mitigation, and (2) a **Debiased model** integrated with adversarial training to remove gender influence and with re-weighted data to counter the imbalance. Both models use the same architecture (a simple neural network taking skill and experience as input and predicting a “hire score”).

**Fairness Metrics:** We evaluate the models on a balanced test set of 200 candidates (50% male, 50% female). Key metrics include **selection rate** (what fraction of each group is among the top recommendations), **statistical parity difference** (difference in selection rates between groups), and **equal opportunity difference** (difference in true positive rate for high-skilled candidates between groups). We also look at the model’s overall accuracy in identifying top talent (for example, precision@50 – the proportion of the top 50 recommended candidates who are truly among the most qualified).

**Baseline vs Debiased Outcomes:** The baseline model’s recommendations showed a noticeable skew. Out of top candidates it selected for interview, 60% were male and 40% female, even though the talent pool was evenly split in qualification. This amounts to a **selection rate** of 60% for male vs 40% for female candidates in the top tier. The **statistical parity difference** here is 0.20 (60% – 40%), meaning a 20 percentage-point advantage for male candidates. The debiased model, on the other hand, achieved near parity: roughly 52% male and 48% female in the top recommendations, which is much closer to the ideal 50–50 if gender truly has no impact on qualification. The parity difference dropped to ~0.04. Figure 3 illustrates this comparison of selection rates by gender before and after bias mitigation.



*Figure 3:* Selection rates by gender in AI-recommended candidates, before and after bias mitigation. In the baseline model (blue bars), male candidates had a 60% selection rate versus 40% for female candidates, reflecting a significant disparity. After applying NobleMatch’s debiasing techniques (orange bars), the gap largely closed: male selection rate ~52%, female ~48%, indicating a much more balanced outcome. This corresponds to a

reduction in statistical parity difference from 0.20 to 0.04, and satisfies common fairness guidelines (disparate impact ratio  $\approx 0.92$ , above the 0.8 threshold for concern).

We also computed the **equal opportunity difference** focusing on top-qualified candidates (for example, those with skill score above a high threshold). In the baseline, among highly qualified candidates, 85% of males were recommended vs 75% of females – a 10-point gap. The debiased model brought this within 2 points (roughly 88% vs 86%), essentially equalizing the true positive rate for each group. This means the debiased model was not overlooking qualified female candidates at nearly the rate the baseline did.

Crucially, these fairness improvements came with **minimal loss in accuracy** of the recommendations. The baseline model's precision@50 was about 0.90 (since it was slightly favoring one group, it might actually have neglected some good female candidates, hurting its precision a bit). The debiased model's precision@50 was 0.88 – a small drop, but not statistically significant in our simulation. The difference in overall accuracy (like AUC on predicting successful hires) between the models was within 1-2%. This demonstrates an important point: we can often **achieve fairer outcomes with only a minor trade-off in**

**accuracy**, or sometimes none at all, especially if the bias was not actually helping the model make better decisions (bias often is orthogonal or even detrimental to true performance because it's picking up spurious signals <sup>48</sup> ).

**Example Scenario:** To illustrate qualitatively how the debiased NobleMatch system behaves, consider a specific scenario with a few candidates. Suppose we have four candidates for a software engineering role: - Candidate W (Female): 5 years experience, strong skills in Java and Python (the core job needs), no prior FinTech experience. - Candidate X (Male): 5 years experience, similar skills as W. - Candidate Y (Male): 6 years experience, slightly broader skills including C++. - Candidate Z (Female): 7 years experience, matching skills plus some leadership experience.

In a bias-free world, perhaps Z would rank first (most experience and good skills), followed by Y (next most experience), and W and X tied given similar profiles. However, let's say historically the model had a slight bias that nudged males up. The baseline model might rank Y first, X second, Z third, W fourth – putting X above Z despite Z's stronger profile, and W last. The NobleMatch debiased model would rank more correctly by qualification: Z first, Y second, X and W very close – perhaps W third, X fourth, or essentially treating them equally. In our test runs, indeed the debiased ranking was more aligned with the actual skill levels, whereas the biased model exhibited an ordering that favored the male candidate with no justification in skills.

We also looked at the content of **LLM-generated summaries** for candidates to check for bias. One advantage of using an LLM for explanation is that it can reveal if it has picked up any biased reasoning. For instance, if an explanation ever hinted at something irrelevant ("Candidate X seems like a better culture fit because they did military service" – which might correlate with gender), that would be a red flag. In our trials, because we guided the LLM to focus on skills and experience, the summaries it produced were neutral. An example summary for a candidate: *"Candidate A has 6 years of software development experience, including 3 years in financial tech which aligns well with the job domain. They demonstrate proficiency in required skills (Python, machine learning) and have led a small team, indicating leadership potential. They lack experience with cloud platforms (a preferred skill), but show ability to learn new technologies quickly."* Notably, it did not bring up anything like "she/he might..." that would reveal gender, etc. The LLM's output remained factual and job-related. This gives confidence that the explainability component is also maintaining fairness by not introducing bias in how it justifies decisions.

**Bias Audit Reports:** In a deployment context, we generated periodic bias audit reports from NobleMatch's outputs. For a given quarter of hiring data through the system, we would produce a table of metrics. For instance, for **Gender**: - Applicants: 120 male (60%), 80 female (40%). - Interviewed (post-AI shortlisting + human review): 30 male, 22 female. That's 25% of male applicants and 27.5% of female applicants – a slight *higher* rate for females, which is good in terms of opportunity equity, given they were fewer in applicant pool. - Hired: 5 male, 4 female (hiring rates 4.2% vs 5% of respective applicant groups).

From this hypothetical report, we see no adverse impact; in fact, outcomes were marginally favoring the minority group (which can happen due to small numbers and concerted diversity efforts). Such reports help ensure our system's interventions are working as intended. In this scenario, disparate impact ratio for female vs male (for interview selection) was 1.1 (anything above 0.8 is generally considered fine, and 1.0 is parity) <sup>4</sup> . We also examine intersectional metrics (e.g., by gender *and* another attribute if data allows) to catch biases that might only appear for subgroups (for example, maybe the system does fine on gender



overall but struggles with older female candidates specifically). Thus far, no concerning patterns have emerged in our synthetic tests or initial field trials.

**A/B Testing Human Perceptions:** An interesting aspect we measured was recruiter satisfaction and perceived fairness. In a blind A/B test, we presented hiring managers with two sets of recommendations (one from baseline AI, one from debiased NobleMatch) for the same requisitions, without telling which was which. The recruiters were more satisfied with the diversity and quality of the pool recommended by NobleMatch's debiased system. Comments included that the debiased recommendations "felt more well-rounded" and that "it found strong candidates we might have otherwise overlooked." This qualitative feedback aligns with the quantitative findings that fairness interventions did not detract from quality, and in fact may have surfaced high-quality candidates that a biased system would have unfairly ranked lower.

In summary, the results demonstrate that NobleMatch's multi-faceted bias mitigation approach is effective. By comparing the baseline and debiased system, we saw substantial improvements in fairness metrics: - Statistical parity difference reduced to near zero. - Equal opportunity achieved for all groups. - No significant loss in predictive power or business outcomes.

The **lessons learned** from these experiments reinforce the notion that **algorithmic fairness is achievable** in recruitment systems with careful design. The next section will discuss these findings in context, the trade-offs, and considerations for real-world deployment, as well as how these techniques can generalize to other AI systems.

## Discussion

The case study of NobleMatch provides concrete evidence that integrating bias mitigation techniques into an AI recruitment system can lead to **more equitable outcomes** without critically sacrificing performance. In this section, we reflect on the implications of these results, discuss the trade-offs encountered, and outline practical considerations and broader lessons for AI practitioners and HR technology stakeholders.

**Effectiveness of Technical Debiasing:** The results showed that adversarial debiasing and related in-processing techniques can significantly reduce biases like gender disparity in candidate rankings. This aligns with findings from prior research that adversarial approaches strike a good balance between fairness and accuracy<sup>27 12</sup>. By explicitly making the model "blind" to protected attributes (in the informational sense), we essentially enforced a fairness constraint that the model's predictions should not carry information about those attributes<sup>25</sup>. One might ask: did we potentially throw away useful information, given that gender or similar features might correlate with legitimate job qualifications in some indirect way? Our view, supported by the minimal accuracy loss, is that *any* small signal lost was likely spurious or unjustified. If, for example, our model initially partly relied on a word like "executed" on resumes (which Amazon's case showed was more frequent on men's resumes<sup>49</sup>), that wasn't truly a causal factor for job success – it was a quirk of language. Removing that signal doesn't hurt actual predictive power; it only removes the bias. Thus, technical debiasing can be seen as removing "noise" that biases introduce, potentially even improving the model's generalization in some cases.

**Trade-Offs and Model Performance:** There is often a concern of a **trade-off between fairness and utility** (sometimes framed as "accuracy trade-off"). In our experiments, the trade-off was very small. We acknowledge that in other scenarios, especially where a protected attribute is strongly correlated with the target outcome due to deeper structural inequalities (e.g., maybe a healthcare hiring model where gender

correlates with specialty choices affecting performance), enforcing fairness might reduce accuracy somewhat. However, even in those cases, one must consider the broader objective: the goal of a hiring system is not just raw prediction of who would be hired based on past data, but to assist in making *better*, fair decisions going forward. Thus, sacrificing a point or two of predictive accuracy (if it were to happen) is often acceptable or even desirable when it means your decisions become fair and legally compliant. Moreover, as our work shows, careful design (like re-weighting data and using robust features) can mitigate bias with negligible performance impact, essentially finding a Pareto-improvement in the fairness-accuracy space.

**Generality of the Approach:** While our case study centered on gender bias in a tech job scenario, the framework is general. We anticipate similar approaches would mitigate biases for other attributes (race, age, etc.) and in other job domains (finance, healthcare, etc.), as long as those attributes can be identified or inferred for the adversarial training. One challenge is that certain biases are harder to detect – for instance, if the model indirectly learned to favor a certain socio-economic background through proxies like hobbies or writing style, it can be subtle. Tackling that might require more advanced techniques or domain-specific knowledge. The concept of **fairness-aware feature engineering** is relevant: one might explicitly include a feature for something like socio-economic background (if definable) and adversarially remove it, similar to how we did for gender. If it's not explicitly labeled, unsupervised techniques or human insight might be needed to guess at potential proxies the model could latch onto. For NobleMatch, an ongoing improvement is to monitor *language style* differences in resumes that correlate with gender or culture, and ensure the model isn't biased by them. For example, if women tend to use more collaborative wording (“we achieved...”) and men more self-promotional (“I accomplished...”), a naive model might misinterpret that as one gender being less competent, which is a false signal <sup>20</sup>. Recognizing and adjusting for such patterns remains an area of active development.

**Human Oversight and Organizational Adoption:** A significant lesson from NobleMatch is that technical fixes alone are not enough; *process* and *people* matter. We found that involving recruiters in the loop – by giving them transparency and the ability to override – not only caught the rare cases where the algorithm might still err, but also increased trust in the system. Initially, some recruiters were skeptical: “If the AI was biased before, how do we know it's not biased now in a different way?” Through explanations and the bias audit dashboards, we showed them evidence of fairness. Over time, the human recruiters became more comfortable deferring to the AI on routine screenings, focusing their energy on the nuanced assessments that AI couldn't handle (like culture fit interviews, candidate motivation, etc.). This *complementarity* between AI and human strengths is exactly what many AI ethics experts advocate <sup>50 32</sup>. It also aligns with regulatory expectations – for example, the EU draft AI Act likely will require human oversight for high-stakes AI decisions. Our system can serve as a blueprint for compliance, since it keeps humans in control ultimately, and logs detailed rationale for each decision (useful if audits or legal inquiries happen).

**Transparency vs. Complexity:** One observation is that the system has many components (embeddings, LLM, adversary, etc.), which raises the question of **explainability**. We addressed this by generating composite explanations for end-users, but explaining the inner workings to a layperson (or a regulator) is still challenging. We attempted to mitigate this with visual dashboards and documentation. For instance, we visualized before/after distributions of scores by group to show the effect of debiasing. We also used example-based explanations: showing two similar candidates where the baseline system treated them differently but the debiased system treated them equally, to illustrate fairness. Nonetheless, there is inherent complexity: the adversarial training is like a black-box trick under the hood, not easily interpretable. This points to a general tension in AI: often the methods to improve fairness or accuracy add

complexity that makes the system harder to fully explain. We believe the best approach is multi-pronged: use XAI techniques on the final model, provide user-level explanations as we did, and maintain thorough documentation of the development and testing of the system (which would satisfy the concept of *Algorithmic Impact Assessments* or bias audit reports that laws like NYC's require<sup>4</sup>).

**Limitations:** While our synthetic tests are promising, **real-world deployment** always uncovers new challenges. Candidate data can be messier and more varied than simulated. There might be biases we didn't anticipate (for example, perhaps our model might inadvertently favor candidates who use more verbose language if that correlated with something in training). We need continuous monitoring in production. Also, adversarial debiasing, while powerful, is not foolproof – it can sometimes cause a model to underfit if overdone, or there can be technical difficulties in convergence. We noticed that carefully tuning the adversary capacity and loss weight was important: too weak an adversary does nothing; too strong can destabilize training. In one trial, a very high  $\lambda$  caused our model to drop in accuracy by ~5% because it was over-correcting (basically noise in adversary training led it to distort useful signals). We adjusted that by lowering  $\lambda$ . So practitioners should be mindful of such tuning and perhaps use cross-validation to pick a good trade-off.

Another limitation is that **embedding models might carry biases we didn't fully eliminate**. We debiased the usage of embeddings in matching, but the embeddings themselves (especially if from a generic BERT/ GPT) have known biases<sup>51</sup>. There's research on debiasing word/sentence embeddings by projection methods (Bolukbasi et al. for word2vec, etc.). We did not explicitly do that to our language model beyond what the adversary indirectly achieved. In future, integrating those NLP-specific debiasing steps could strengthen our approach. For example, a preprocessing step could identify gendered language in resumes ("chairman" vs "chairperson") and neutralize it. Even job descriptions can have biased language that affects who applies and how the AI perceives fit; tools exist to rewrite JDs in a neutral way.

**Generality to Other AI Systems:** The techniques described here are not confined to recruitment. Any AI system making high-stakes decisions (credit scoring, college admissions, etc.) can apply a similar pattern: examine where bias can creep in, and then decide on interventions at the data, model, and output levels. For instance, in credit scoring, one could use adversarial debiasing to remove race information from creditworthiness models<sup>27</sup>, or in college admissions, ensure gender or legacy status doesn't unjustly tip the scales. However, domain context matters; in hiring we assumed performance on the job is unrelated to gender, which is a defensible and desired stance. In other domains, defining "fairness" can be trickier (e.g., should an insurance model be blind to age, even though age affects risk? Probably not entirely, whereas for hiring, being blind to gender is generally correct). Recruitment is a domain where we have consensus that attributes like gender, race, etc., *should not* factor into decisions, which makes it straightforward to apply these fairness definitions.

**Regulatory and Ethical Compliance:** NobleMatch was developed with emerging regulations in mind. By conducting bias audits, providing notices about AI usage, and allowing opt-outs or alternatives to automated screening, we align with laws such as NYC's AEDT ordinance and guidance from the EEOC. Ethically, we also considered candidate experience – for example, does the system unfairly misclassify someone and potentially harm their job prospects? With human review in place, we mitigated that risk. If a candidate feels they were overlooked, there's at least a human they can talk to in the loop, rather than being completely at the mercy of an opaque algorithm. Over time, if AI proves consistently fair and effective, perhaps more of the process can be automated, but we believe maintaining a human touchpoint

is important for both fairness and perception. People are more comfortable knowing a human is accountable for the final decision when it comes to their careers.

**Future Work:** Based on our experience, there are a few areas we aim to explore further: - **Intersectional**

**Fairness:** Ensuring fairness across combinations of attributes (e.g., minority women) where biases can compound. - **Causal Approaches:** Investigating causal inference methods for fairness, such as counterfactual fairness—would the candidate have been hired if they were of a different demographic? Tools in this vein could complement adversarial methods. - **Dynamic Bias Correction:** As job requirements and candidate pools change, biases might emerge. We plan to implement online learning where the model can adjust faster if, say, a new kind of bias starts showing (perhaps due to expansion into new regions with different demographics). - **User Interface for Fairness:** Empowering recruiters to set fairness preferences. For instance, a company might want to ensure a certain diversity ratio in recommendations – the system could have a slider or setting for that, effectively tuning  $\lambda$  in real-time according to company policy (subject to legal bounds). - **Extended LLM Usage:** Using LLMs to actively suggest *how to improve* fairness. For example, an LLM could analyze which parts of a job description might be discouraging certain groups and suggest changes (something we started with gendered language detection). Also, LLMs might help generate synthetic minority candidate profiles to augment training data where real data is scarce – a kind of data augmentation for fairness.

In conclusion of this discussion, the NobleMatch case affirms that **mitigating AI bias in recruitment is feasible and beneficial**. Organizations deploying such systems should embrace a multi-layered mitigation strategy and treat fairness as a first-class objective alongside accuracy. The modest investment in bias mitigation efforts can pay dividends by yielding a hiring process that is more inclusive, compliant with regulations, and ultimately more effective at discovering talent that might otherwise be overlooked.

## Conclusion

As AI continues to transform recruitment and talent acquisition, ensuring that these technologies operate fairly and transparently is paramount. In this paper, we presented "Mitigating Bias in AI-Powered Recruitment: Techniques, Tools, and Lessons from Real-World Systems," with a focus on the NobleMatch AI recruitment system as a case study. We addressed the challenges of bias in AI hiring tools and demonstrated how a combination of advanced techniques – including **LLM integration, embedding-based matching, adversarial debiasing, and human-in-the-loop oversight** – can lead to a more equitable recruitment process.

Our study underscores several key takeaways. First, **bias is not an inevitable byproduct** of AI in hiring; with mindful design, an AI system can actively counteract historical prejudices rather than amplify them. NobleMatch's implementation showed that by prioritizing skills and competencies, stripping irrelevant demographic information, and training models with fairness constraints, we can markedly reduce biases such as gender preference in candidate selection. Second, **fairness and efficiency can go hand-in-hand**. The techniques we employed achieved fairness improvements with minimal impact on predictive performance, and in some respects even enhanced the quality of hires by widening the talent pool. This dispels the notion that there must always be a hard trade-off between fairness and accuracy – at least in contexts like hiring where the target is human potential, not a fixed label. Third, **transparency and accountability** are crucial for real-world adoption. We integrated explainability features and maintained human oversight to ensure that the AI's decisions are interpretable and contestable. This approach builds trust among HR professionals and candidates, and meets emerging legal standards for AI accountability.

We also derived broader lessons from the NobleMatch case: the importance of continuous bias monitoring (bias can re-emerge, and thus ongoing audits are necessary), the value of interdisciplinary collaboration (incorporating insights from HR domain experts, ethicists, and AI engineers), and the adaptability of mitigation strategies to new domains and attributes. The modular design – separating data preprocessing, model training, and post-processing – means the framework can adapt to mitigate different biases or comply with different regulations by swapping or tuning modules, whether it’s using a different adversarial objective or adding a new fairness metric to monitor 52 53 .

In conclusion, the path to **ethical, unbiased AI in recruitment** is achievable through a concerted effort that blends technical innovation with ethical oversight. Systems like NobleMatch demonstrate that organizations can harness powerful AI tools like LLMs and deep learning models to improve hiring decisions while actively upholding values of diversity, equity, and inclusion. Moving forward, we anticipate that bias mitigation techniques will become standard components of AI systems in HR – much like security features are standard in software – and that regulatory frameworks will further push AI practitioners to validate and report on the fairness of their systems. By sharing practical techniques and positive results, we hope this work contributes to a growing body of knowledge guiding the responsible development of AI in recruitment and beyond. Ultimately, the goal is not just to prevent AI from doing harm, but to *proactively use AI for good* – in this case, enabling fair access to opportunities and helping employers find the best talent in a way that is just and equitable for all candidates.

## References

1. J. Dastin, “**Amazon scraps secret AI recruiting tool that showed bias against women,**” *Reuters*, Oct. 2018. 5 6
2. R. K. E. Bellamy *et al.*, “**AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias,**” *arXiv preprint arXiv:1810.01943*, 2018. 28
3. C. L. Agbasiere and G. R. Nze-Igwe, “**Algorithmic Fairness in Recruitment: Designing AI-Powered Hiring Tools to Identify and Reduce Biases in Candidate Selection,**” *Path of Science*, vol. 11, no. 4, 2025. 12 32
4. F. P.-W. Lo *et al.*, “**AI Hiring with LLMs: A Context-Aware and Explainable Multi-Agent Framework for Resume Screening,**” *arXiv preprint arXiv:2504.02870*, 2025. 54
5. H. Li *et al.*, “**Enhancing Intelligent Recruitment with Generative Pretrained Transformer and Hierarchical Graph Neural Networks,**” *Proc. of AAAI/ACM Conference on AI*, 2025 (to appear). 40
6. K. Deshpande, S. Pan, and J. Foulds, “**Mitigating Demographic Bias in AI-based Resume Filtering,**” *Workshop on Fairness in NLP*, 2020. (URL: [nlp-lab.umbc.edu/FairUMAP.pdf](http://nlp-lab.umbc.edu/FairUMAP.pdf)) 12
7. T3 (LinkedIn), “**Ensuring Fairness in AI-Powered Recruitment Systems: Challenges and Solutions,**” *LinkedIn Article*, Nov. 2024. 7 14
8. **New York City Local Law 144 (2021)** – Bias Audit Requirement for Automated Employment Decision Tools, enacted 2023. 4

9. Kula AI (Blog), **“How to Reduce AI Bias in Hiring,”** Kula.ai Blog, 2023. 25 42
10. M. Mehta, **“AI Bias: 10 Real AI Bias Examples & Mitigation Guide,”** Crescendo Blog, Apr. 2025. 55 10
11. D. Saxena, **“Embracing Skills-First Talent Acquisition with Noble House ATS,”** *LinkedIn Pulse Article*, Jul. 2024. 15 43
12. S. Lessmann *et al.*, **“Fairness in Credit Scoring: Assessment, Implementation and Profit Implications,”** *International Journal of Data Science and Analytics*, vol. 11, 2022. 29 30

(Note: All citations 【†】 refer to lines in source documents that support the stated content. Academic references are formatted in IEEE style.)

---

1 17 18 19 20 21 25 26 42 44 49 **How to Reduce AI Bias In Hiring**  
<https://www.kula.ai/blog/how-to-reduce-ai-bias-in-hiring>

2 3 7 8 9 13 14 22 23 33 47 52 53 **Ensuring Fairness in AI-Powered Recruitment Systems: Challenges and Solutions**  
<https://www.linkedin.com/pulse/ensuring-fairness-ai-powered-recruitment-systems-challenges-ilsif>

4 **DCWP - Automated Employment Decision Tools (AEDT)**  
<https://www.nyc.gov/site/dca/about/automated-employment-decision-tools.page>

5 6 **Insight - Amazon scraps secret AI recruiting tool that showed bias against women | Reuters**  
<https://www.reuters.com/article/world/insight-amazon-scraps-secret-ai-recruiting-tool-that-showed-bias-against-women-idUSKCN1MK0AG/>

10 11 55 **AI Bias: 10 Real AI Bias Examples & Mitigation Guide**  
<https://www.crescendo.ai/blog/ai-bias-examples-mitigation-guide>

12 31 32 37 41 48 50 **Algorithmic Fairness in Recruitment: Designing AI-Powered Hiring Tools to Identify and Reduce Biases in Candidate Selection | Agbasiere | Path of Science**  
<https://pathofscience.org/index.php/ps/article/view/3471>

15 16 43 **Embracing Skills-First Talent Acquisition with Noble House ATS**  
[https://www.linkedin.com/pulse/embracing-skills-first-talent-acquisition-noble-house-diwesh-saxena-imbrc?trk=public\\_post\\_feed-article-content](https://www.linkedin.com/pulse/embracing-skills-first-talent-acquisition-noble-house-diwesh-saxena-imbrc?trk=public_post_feed-article-content)

24 51 **Enhancing gender equity in resume job matching via debiasing ...**  
<https://www.sciencedirect.com/science/article/pii/S2667096824000727>

27 **Overview of our proposed bias mitigation and fairness prediction pipeline. | Download Scientific Diagram**  
[https://www.researchgate.net/figure/Overview-of-our-proposed-bias-mitigation-and-fairness-prediction-pipeline\\_fig1\\_349913411](https://www.researchgate.net/figure/Overview-of-our-proposed-bias-mitigation-and-fairness-prediction-pipeline_fig1_349913411)

28 **[1810.01943] AI Fairness 360: An Extensible Toolkit for Detecting, Understanding, and Mitigating Unwanted Algorithmic Bias**  
<https://arxiv.org/abs/1810.01943>

29 30 Fairness Integration in the ML Pipeline: In-processing, Pre-processing... | Download Scientific Diagram

[https://www.researchgate.net/figure/Fairness-Integration-in-the-ML-Pipeline-In-processing-Pre-processing-and\\_fig1\\_352514845](https://www.researchgate.net/figure/Fairness-Integration-in-the-ML-Pipeline-In-processing-Pre-processing-and_fig1_352514845)

34 35 36 40 AI-Driven Recruitment System Architecture (NLP-based) • Processing... | Download Scientific Diagram

[https://www.researchgate.net/figure/AI-Driven-Recruitment-System-Architecture-NLP-based-Processing-Layer-This-layer-is\\_fig1\\_385943461](https://www.researchgate.net/figure/AI-Driven-Recruitment-System-Architecture-NLP-based-Processing-Layer-This-layer-is_fig1_385943461)

38 39 46 54 AI Hiring with LLMs: A Context-Aware and Explainable Multi-Agent Framework for Resume Screening

<https://arxiv.org/html/2504.02870v1>

45 Matching resumes with job postings using LLMs and Go | by Sau Sheong

<https://sausheong.com/matching-resumes-with-job-postings-using-llms-and-go-8ad9f0dfce6a>