Resume seniority classification under style and demographic bias

RESEARCH GenAI

by **E. Fabrikant and R. Deeb**

Abstract

The Complexity

Inferring a candidate's seniority (Junior, Mid, Senior) is notoriously difficult due to inconsistent resume formatting, enormous variations in content length, and the lack of standardized terminology across industries.

Self-Presentation

Candidates often employ strategies to "game" the system. Overstating experience (inflation) or understating qualifications (humility) creates noise that traditional keyword-based models struggle to interpret correctly.

Introduction

Research Problem

WHAT I WANT TO SOLVE

How do modern LLMs, evaluated via zero-shot inference for large foundation models and fine-tuning for smaller specialized models, interpret candidate seniority from resumes in cases of inflation or self-minimization, and do the seniority-assessment and demographic biases (gender, ethnicity) documented in earlier research still appear in today's models?

Questions

INVESTIGATION QUESTIONS

·Accuracy and performance: the difference between zero-shot models, baseline, and fine-tuned models.

·Seniority bias under style manipulation: Do models know that you overstated and understated?

·Demographic bias: Do social bias still exist?

Related Literature

1) Reading between the lines: classifying resume seniority with large language models
2) A Gemini Pro Vision based framework for resume–job description matching
3) Bias and fairness in large language models for job–resume matching
4) When your resume is (not) turning you down: modelling ethnic bias in resume screening
5) Measuring gender and racial biases in large language models: intersectional evidence from automated resume evaluation
6) Evaluating LLM behavior in hiring: implicit weights, fairness across groups, and alignment with human preferences
7) Hidden in plain sight: evaluation of the deception detection capabilities of LLMs in multimodal settings
8) Effects of written self-promotion on gender bias and decision quality

FEASIBILITY + FINE-TUNING ADVANTAGE

REFERRED ARTICLES: [1] , [7]

- Seniority classification from resumes is doable
- Writing style (inflation vs modesty) can bias predictions
- Research shows dedicated fine-tuning often beats zero-shot

FAIRNESS NEEDS CONTROLLED TESTING

REFERRED ARTICLES: [3] , [4], [5]

- HR LLM systems can stay biased even with high performance
- Signals like institution prestige can skew outcomes
- Intersectional patterns exist. Fairness tests are mandatory, not just accuracy

WHY MODELS “FALL” UNDER STYLE SHIFTS

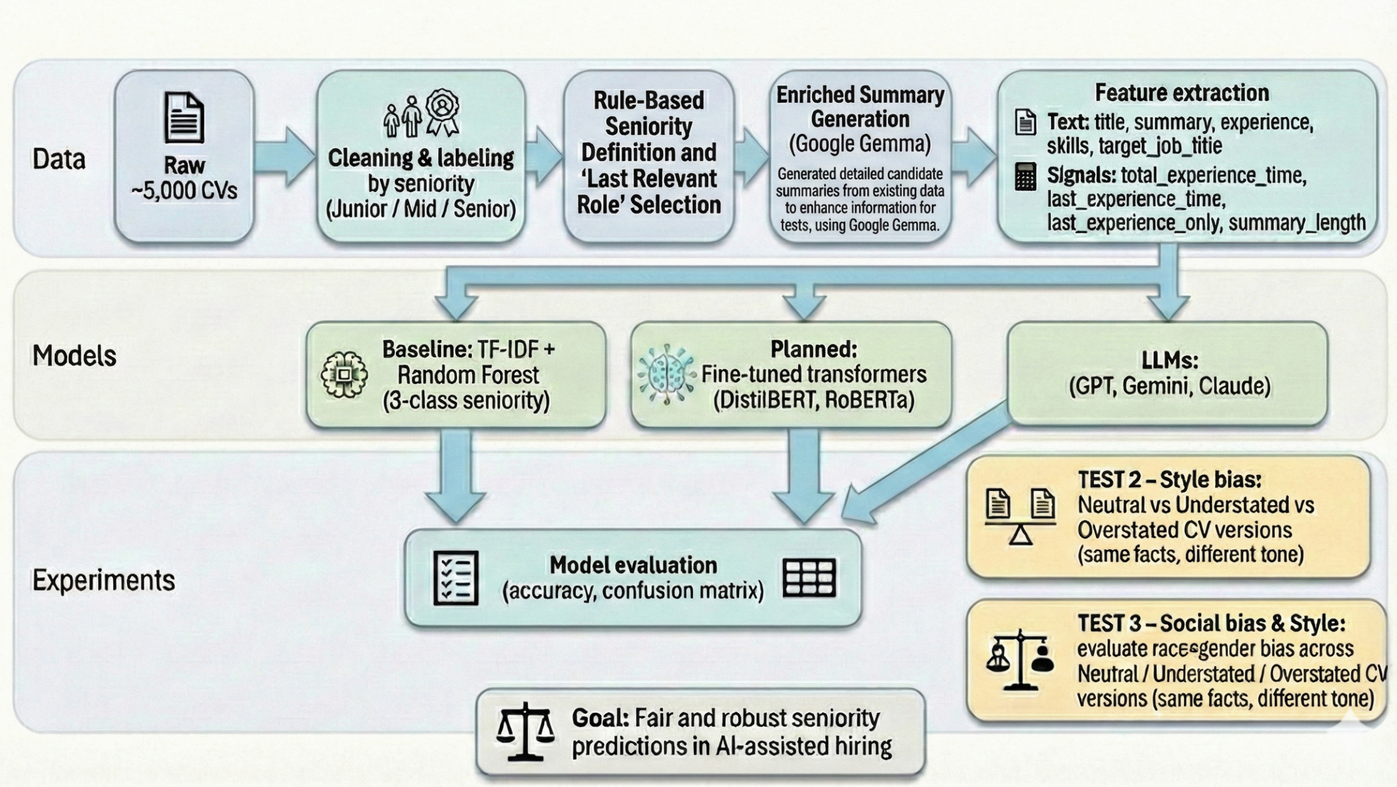
REFERRED ARTICLES: [1] , [6] , [8]

- Models may overreact to self-glorification cues
- They can overweight titles/keywords instead of real evidence
- The Triplets test (Neutral/Over/Under) is critical to expose this

WHAT WE TEST IN THIS PROJECT

- Not only “who is most accurate”
- Stability across Neutral vs Overstated vs Understated resumes
- Bias checks, including combined attributes
- Word/phrase sensitivity: what pushes Junior vs Mid vs Senior decisions

Methodology



Dataset Overview master_resumes.jsonl from HuggingFace

DATASET SIZE : 4,817 CANDIDATE PROFILES

- Source: HuggingFace master_resumes.jsonl (N = 4,817)
- One row = one candidate profile (complete resume record)
- Free-text fields: summary + role descriptions (self-presentation signal)
- Structured sections (JSON lists): experience, skills, education, projects
- Key label signal: each experience entry includes a level attribute (used later for 3-class mapping)

PROFILE STRUCTURE

Profile info: id/name, location, preferences, links

Summary (Text)

Experience (list): {Title, Company, Dates, Description, Level}

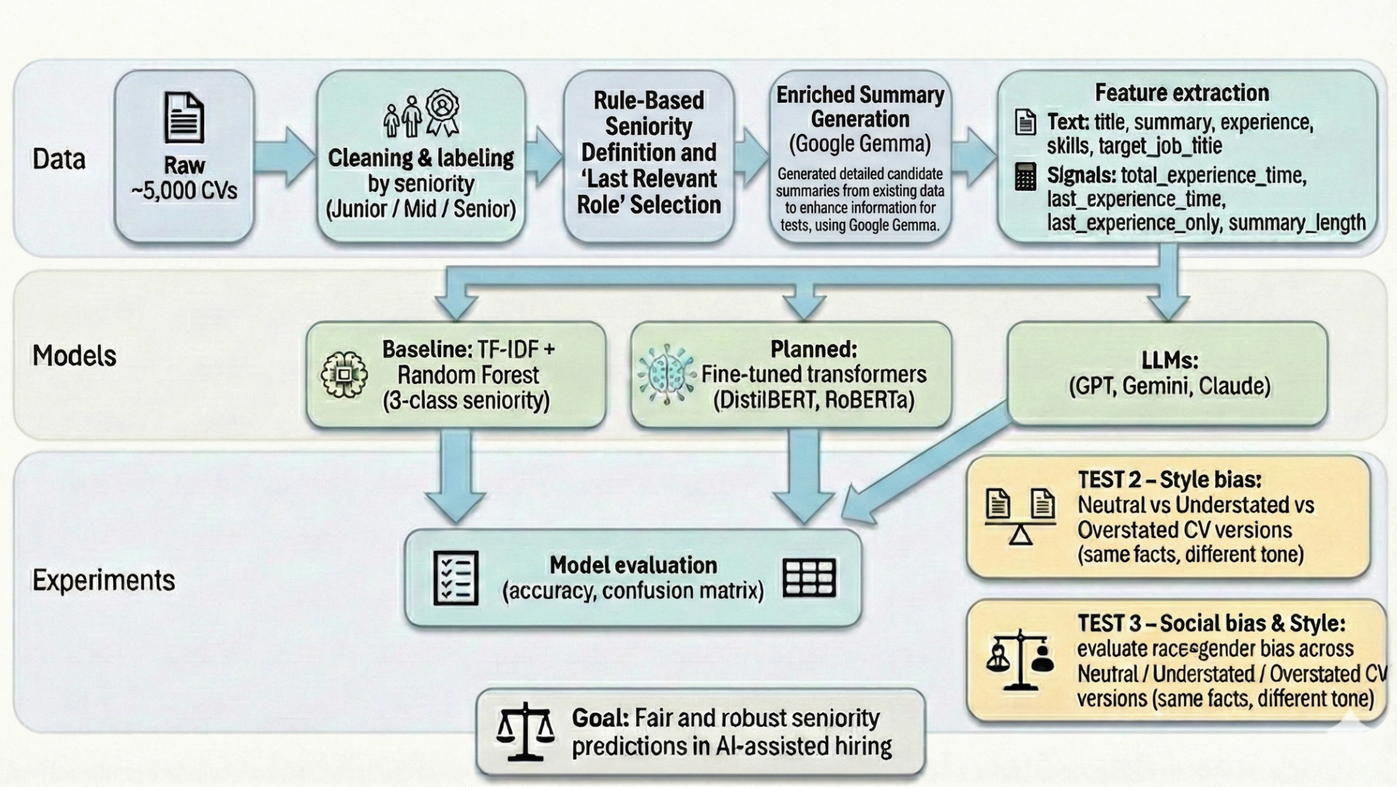
Skills (lists)

Education / Certifications

Projects / Achievements

Teaching / Workshops

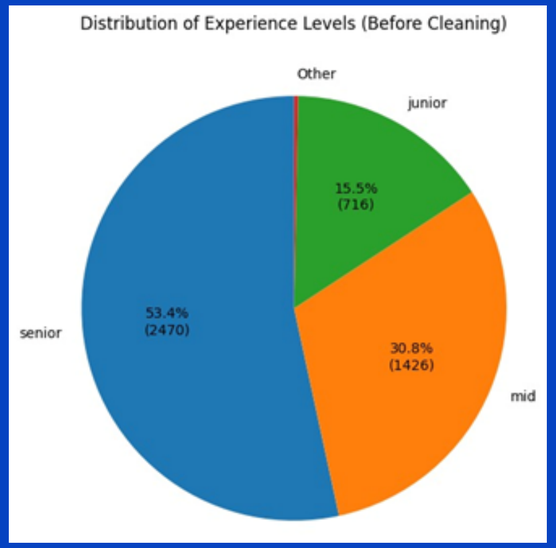
Methodology



Cleaning & Labeling (3–Class Seniority)

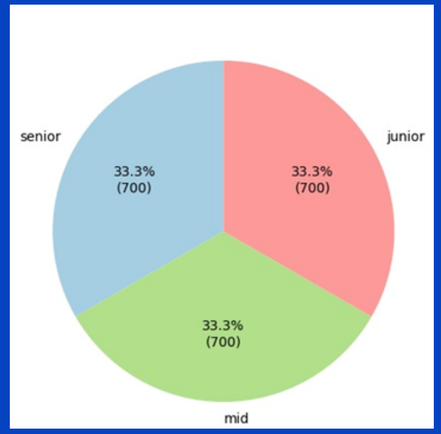
- Ground-truth source: level field inside each experience entry
- Label mapping: collapse original levels → {Junior, Mid, Senior}
- Why cleaning: raw resumes have inconsistent dates, titles, and skill formats
- Imbalance check: original class distribution is skewed

Before cleaning:

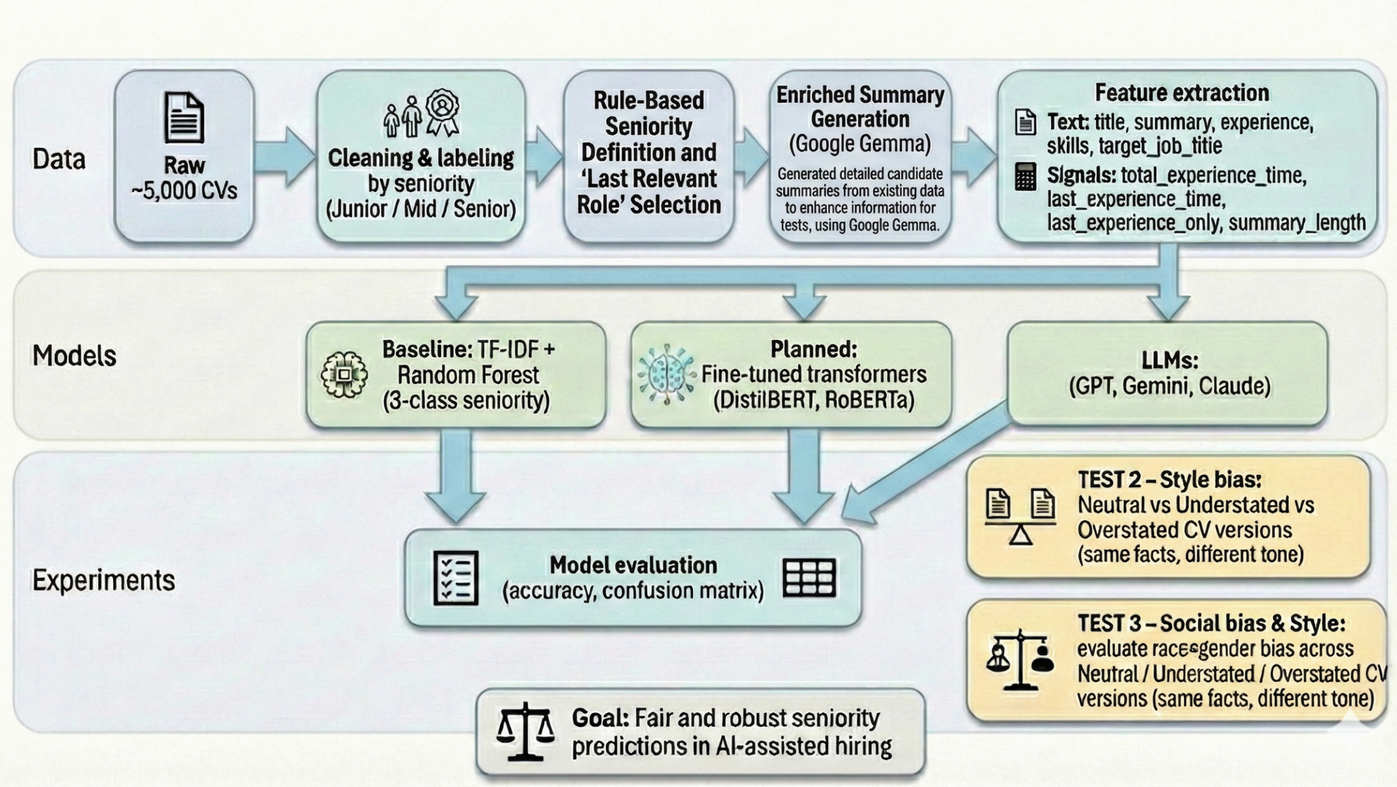


Balanced subset for modeling: build 2,100 resumes (700 per class) for fair comparison

After cleaning:



Methodology



Rule-Based Seniority Target: Last Relevant Role

MAKE EACH CANDIDATE COMPARABLE BY SELECTING ONE CONSISTENT ROLE TARGET

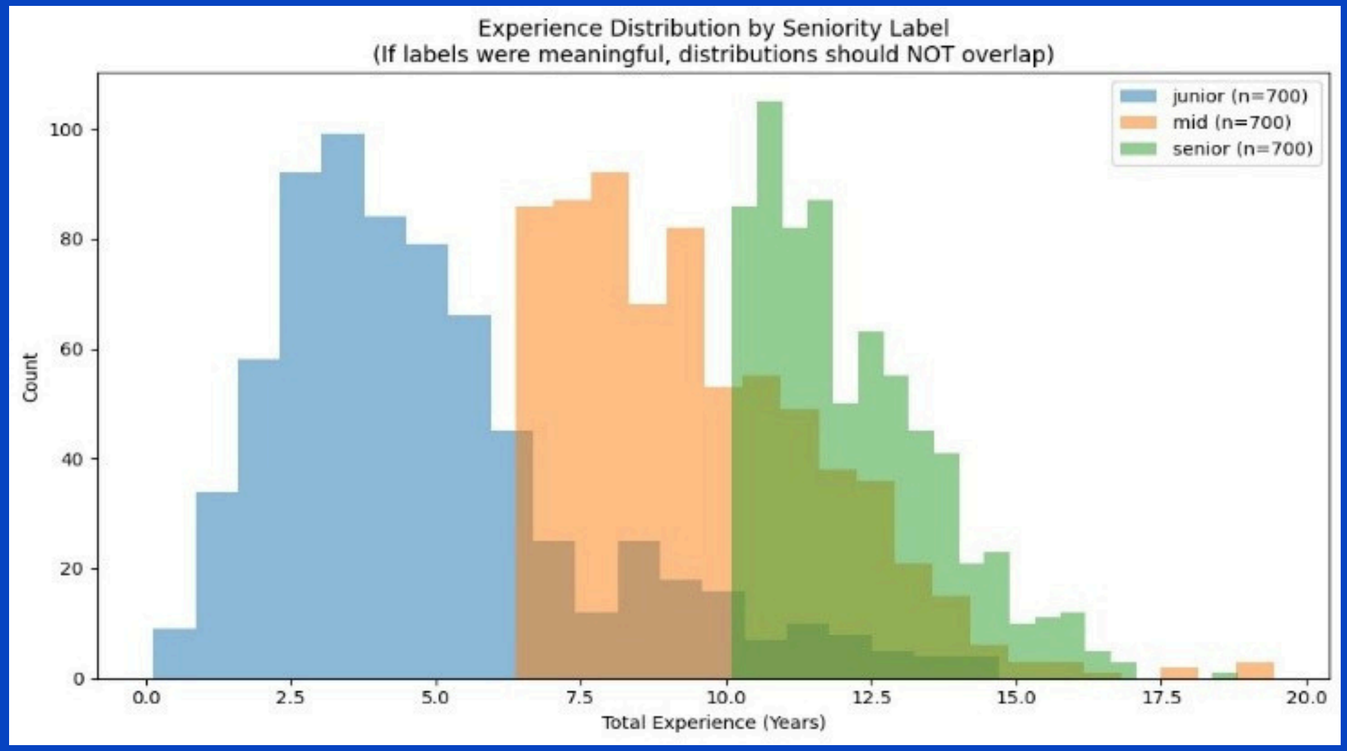
- **PROBLEM:**

  Candidates have multiple roles → need one consistent seniority target

- **SOLUTION:**

  - Single role: use that role as the seniority indicator
  - Multiple roles: select the most recent / ongoing position (“last relevant role”)
  - Anti-inflation constraint: prevent superficial title changes from upgrading seniority
  - Tenure signals: compute total career duration + last-role duration from employment dates

Experience distribution by seniority label

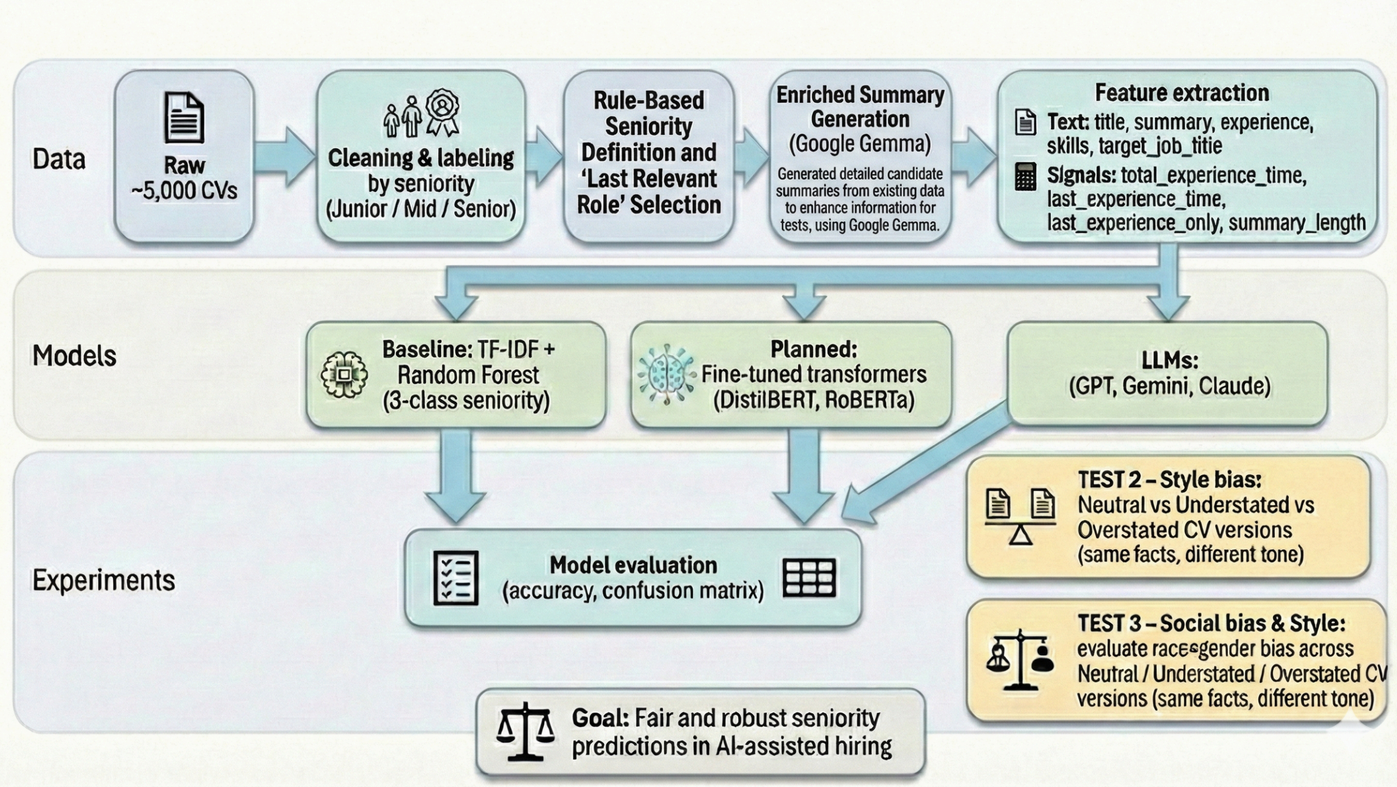


Observation: experience-years overlap across classes.

Text evidence is required too

Overlap shows seniority isn’t determined by tenure alone.

Methodology



Enriched Summary Generation

Metrics:

Word Count Comparison

Whether summaries are a reasonable length

How we Compute:

1- Simply count words
2- Get min and max and average
3- Compare them

Semantic SImularity

how similar the summary is to the resume content

How we Compute:

1- concatenates three key fields
2- Convert Text to Numbers With all-MiniLM-L6-v2 embedding model
3- $$\cos(\mathbf{e}_r^{(i)}, \mathbf{e}_s^{(i)}) = \frac{\mathbf{e}_r^{(i)} \cdot \mathbf{e}_s^{(i)}}{\|\mathbf{e}_r^{(i)}\| \ \|\mathbf{e}_s^{(i)}\|}$$

Metrics results

1) Zety resume profile

Word Count Comparison

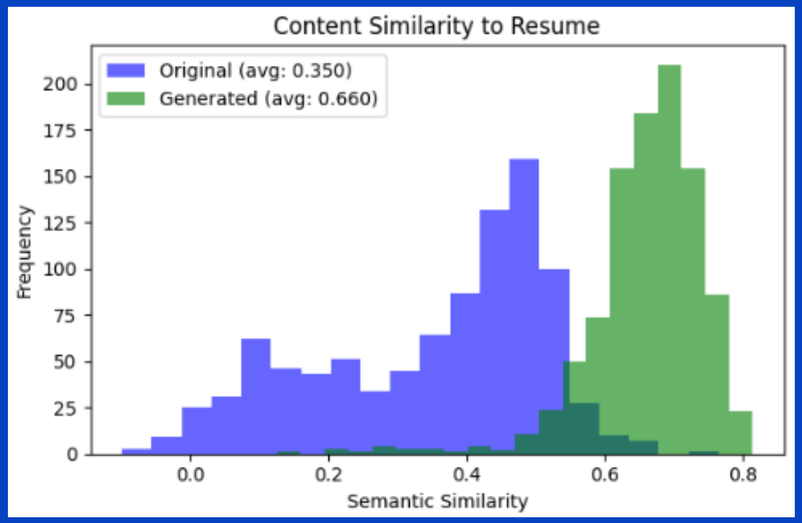
Mistral 3b:

Average	20 words	170 words
Min	1	4
Max	109	280
Too short (<30)	4685	2

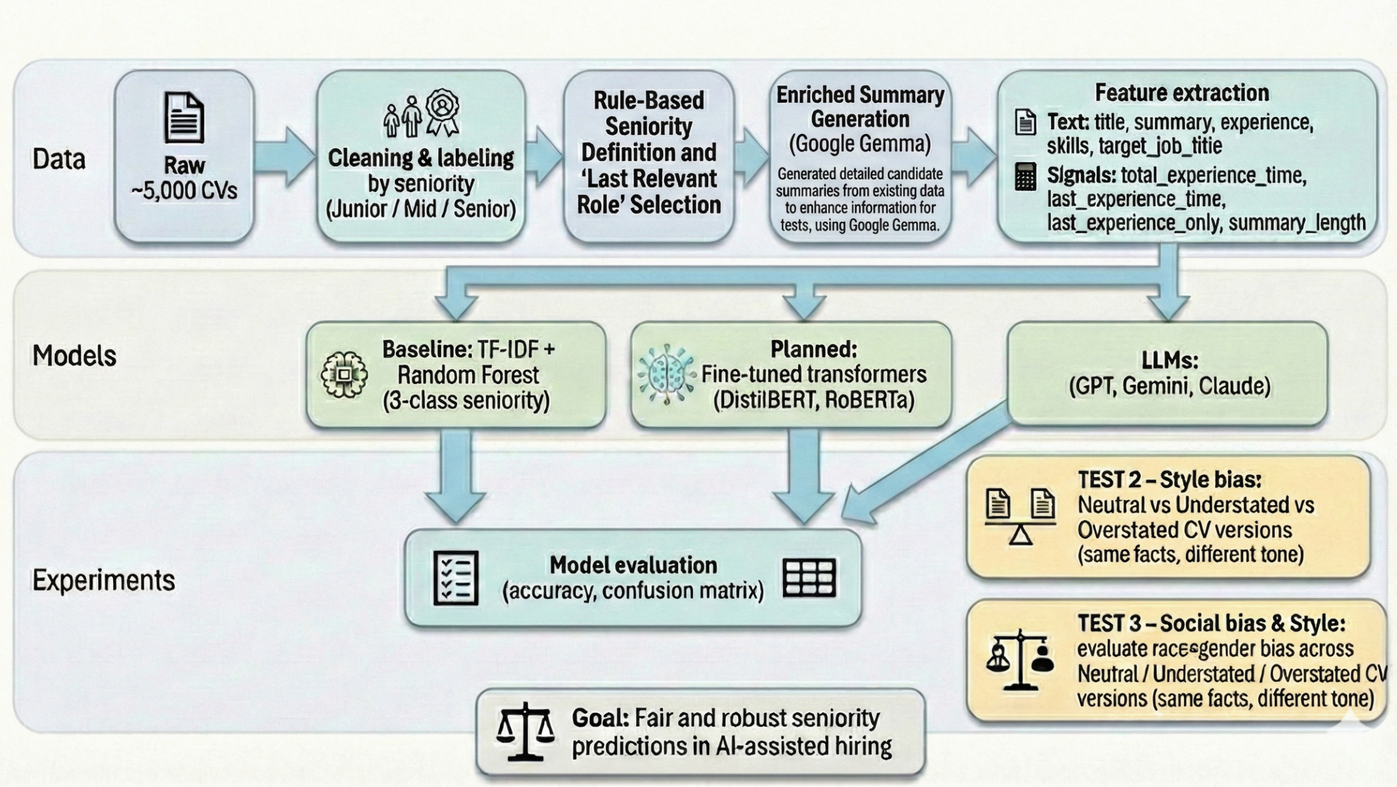
Google Gemma:

Average	20 words	60 words
Min	1	20
Max	109	90
Too short (<30)	4685	2

Semantic simularity



Methodology



Feature Extraction

COMBINE RESUME TEXT WITH SUPPORTING HELP SIGNALS WHILE PREVENTING LABEL LEAKAGE

- Text inputs: job titles • role descriptions • skills • enriched summary
- Supporting help signals: total experience duration • last-role duration • summary length
- Unified input format: keep features consistent across all model families
- Leakage prevention: remove explicit seniority labels from experience entries
- Goal: force models to infer seniority from content, not “read the answer”

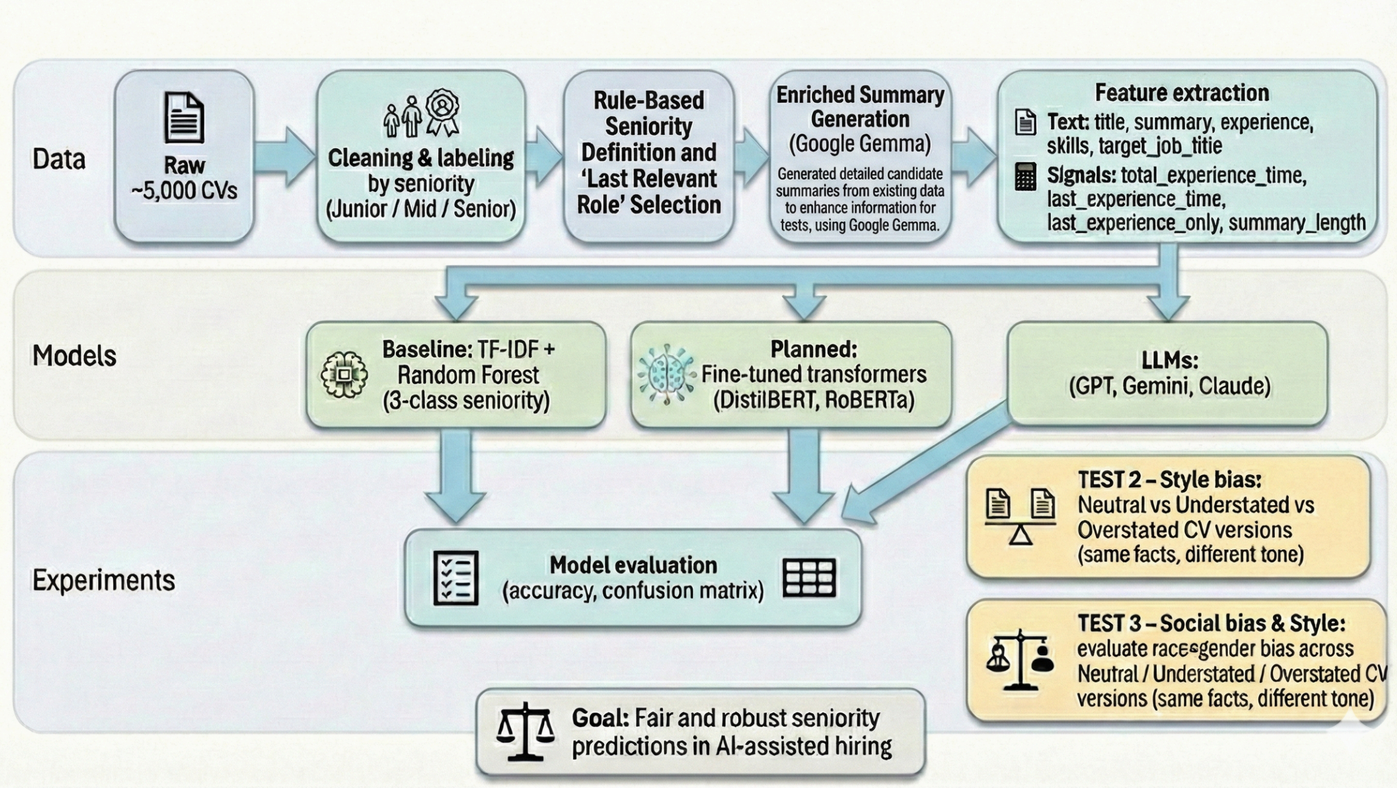
TEXT FEATURES

- Titles
- Role descriptions
- Skills
- Enriched summary

NUMERIC / META SIGNALS

- Total years experience
- Last role duration
- Summary length

Methodology



Baseline: TF–IDF + Random Forest

A fast lexical reference model based on keyword statistics

Representation: TF-IDF converts each resume into a sparse keyword-weight vector

Classifier: Random Forest learns non-linear rules from these lexical features

Why included: interpretable + computationally efficient baseline for comparison

What it captures: common resume phrasing and surface cues

Main limitation: no word order / long-range context → sensitive to synonyms & embellishment

TF-IDF (TERM WEIGHTING)

TF: how often a word appears in this resume
IDF: down-weights words that appear in many resumes (“common words”)

RANDOM FOREST (ENSEMBLE TREES)

Many decision trees vote on the label
Each tree uses different feature subsets
Captures keyword interactions

DistilBERT & RoBERTa (Both BERT–style encoders)

BERT

- **BERT is an encoder-only Transformer that builds contextual token representations using bidirectional self-attention.**
- **For classification, a special CLS token summarizes the sequence and is fed to a small head to predict the label.**

DistilBERT

Compressed BERT → faster, good efficiency baseline

RoBERTa

Reduces reliance on isolated keywords more than TF-IDF
BERT-style with improved pretraining recipe → stronger representations

Both

Both are encoder-only Transformers (good for classification)
Both fine-tuned with the same head and loss for fair comparison

Transformer Resume Encoding (Self–Attention)

EQUATIONS

$$score_{n,m} = q_n^T k_n$$

Resume text is split into sub-word tokens → embeddings

Each token is projected into Query / Key / Value vectors

$$q_n = \beta_q + \Omega_q x_n \quad k_n = \beta_k + \Omega_k x_n \quad v_n = \beta_v + \Omega_v x_n$$

Attention scores measure relevance between tokens →then use of SoftMax

$$a_{n,m} = softmax\left(\frac{score_{n,m}}{\sqrt{d_k}}\right)$$

Output token is a weighted sum of value vectors (context mixing)

$$y_n = \sum_{m=1}^{N} a_{n,m} v_m$$

Multi-head attention captures multiple relationship patterns in parallel

Final CLS embedding summarizes the resume for classification

Fine-Tuning a Transformer Encoder for Classification

Input resume → tokenize → Transformer encoder

Take CLS as resume representation

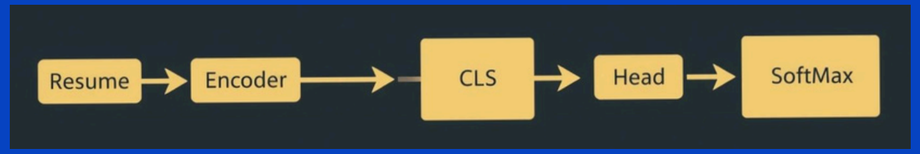
Compute logits: $z = w \times h_{cls} + b$

SoftMax :

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

Train with cross-entropy:

$$L = \frac{1}{n} \sum_{i=1}^{n} -\log \frac{e^{z_{i,y_i}}}{\sum_k e^{z_{i,k}}}$$



Backpropagation Strategy

1. **Gradient Descent**

   Errors are propagated backward to calculate weight influence.

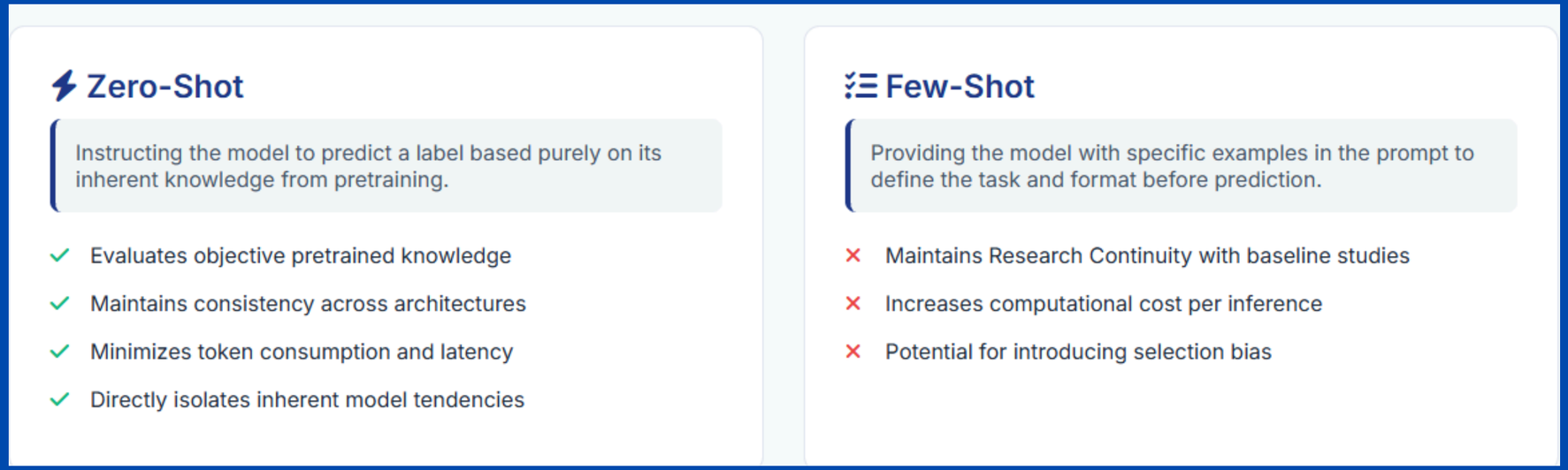
2. **Weight Updates**

   Weights are adjusted iteratively to minimize loss across batches.

3. **Validation Loop**

   The optimal model state is saved based on validation accuracy.

LLMs as Classifiers: Zero-Shot Protocol



- Zero-shot: no examples provided → isolates pretrained seniority reasoning
- Why not few-shot: examples can introduce selection bias and reduce comparability across models
- Strict output constraint: single-token label (no explanations) to keep evaluation consistent
- Compared fairly: same inputs, same labels, same test split as other model families

Foundation LLMs Evaluated

All used in the same zero-shot protocol and the same label space {Junior, Mid, Senior}

GPT-5 (OpenAI): instruction-tuned foundation model used as a zero-shot classifier for seniority labels.

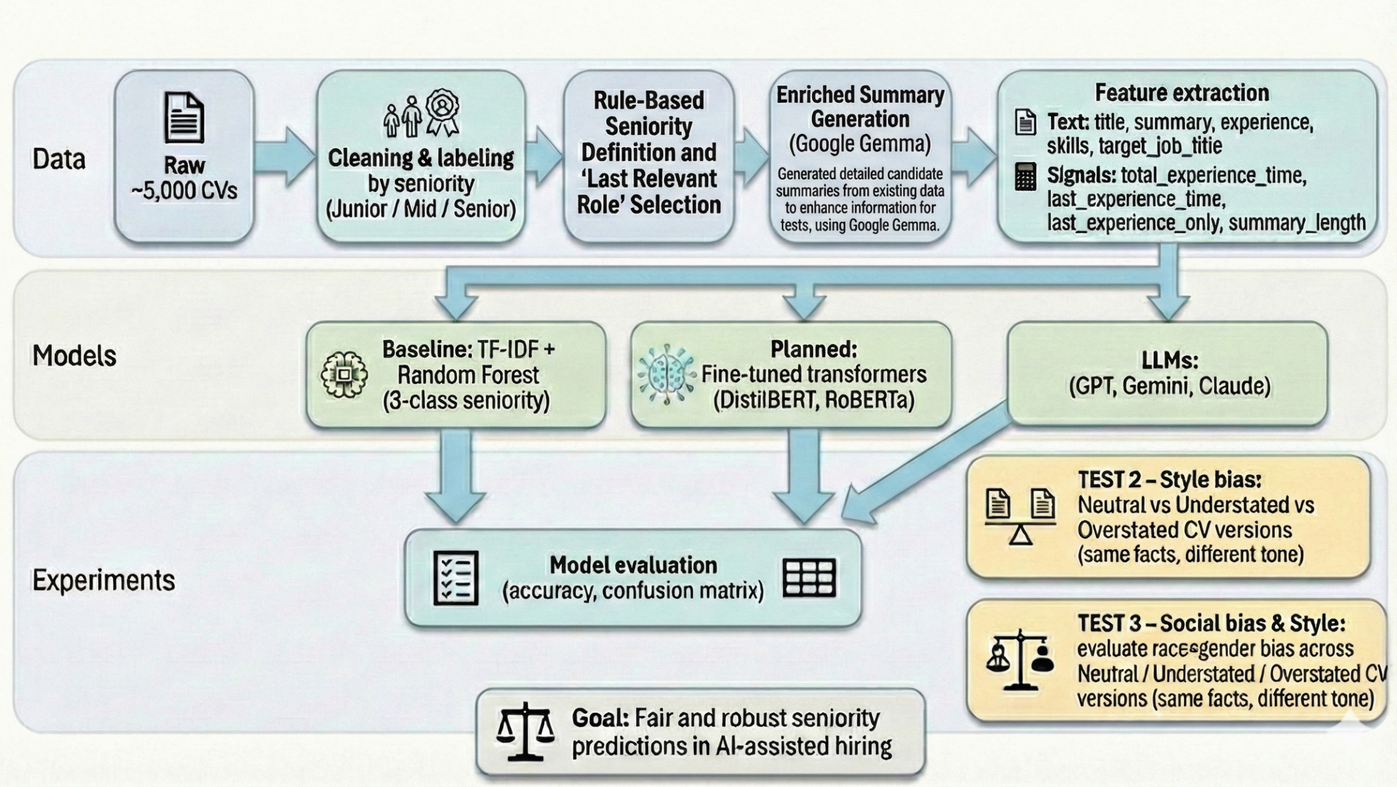
Gemini 3 Pro (Google): multimodal-capable foundation model; evaluated here in text-only mode for consistent comparison.

Claude Sonnet 4.5 (Anthropic): instruction-following foundation model; evaluated as a prompt-only classifier under the same constraints.

Why these three: strong, widely-used, modern LLM baselines representing different providers.

Control: identical prompt format + strict “one label only” output rule.

Methodology



Model Evaluation Protocol Test 1

Goal: compare models fairly on the same 3–class task (Junior/Mid/Senior)

Same data split: identical train/test/validation partitions for all models

EQUATIONS

$$\text{Recall: } R_k = \frac{M_{kk}}{\sum_{j=1}^{K} M_{kj}}$$

$$\text{Precision: } P_k = \frac{M_{kk}}{\sum_{i=1}^{K} M_{ik}}$$

$$F1_k = 2 \times \frac{P_k \times R_k}{P_k + R_k}$$

Core metrics:

Macro-F1 (balanced across classes)

$$\text{Macro F1} = \frac{1}{K} \sum_{k=1}^{K} F1_k$$

+ Accuracy
Accuracy measures the overall proportion of correct predictions out of the total number of samples (N)

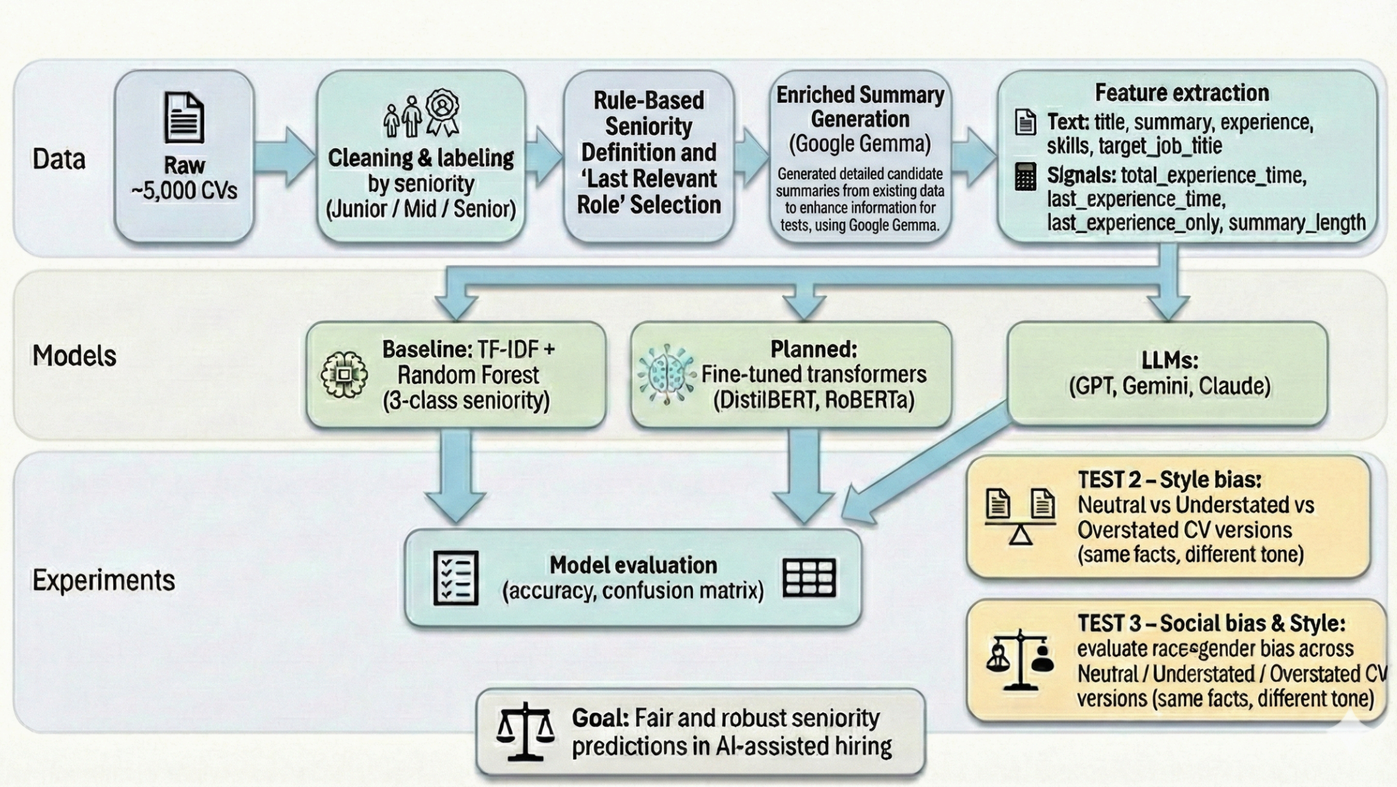
$$Accuracy = \frac{\sum_{k=1}^{K} M_{kk}}{\sum_{i=1}^{K} \sum_{j=1}^{K} M_{ij}}$$

Diagnostics: Confusion Matrix to inspect common misclassifications

Choose the best model for each category - junior F1 , Mid F1 , Senior F1 , Acurracy , Macro-F1 on the held-out test set

For each specific class k.

Methodology



Bias Tests (Controlled Evaluations)

Test 2 - **Style bias**: same candidate, 3 writing styles (Overstated / Neutral / Understated)

Test 3 - **Social bias**: same resume, name-only swap across 4 demographic-coded groups

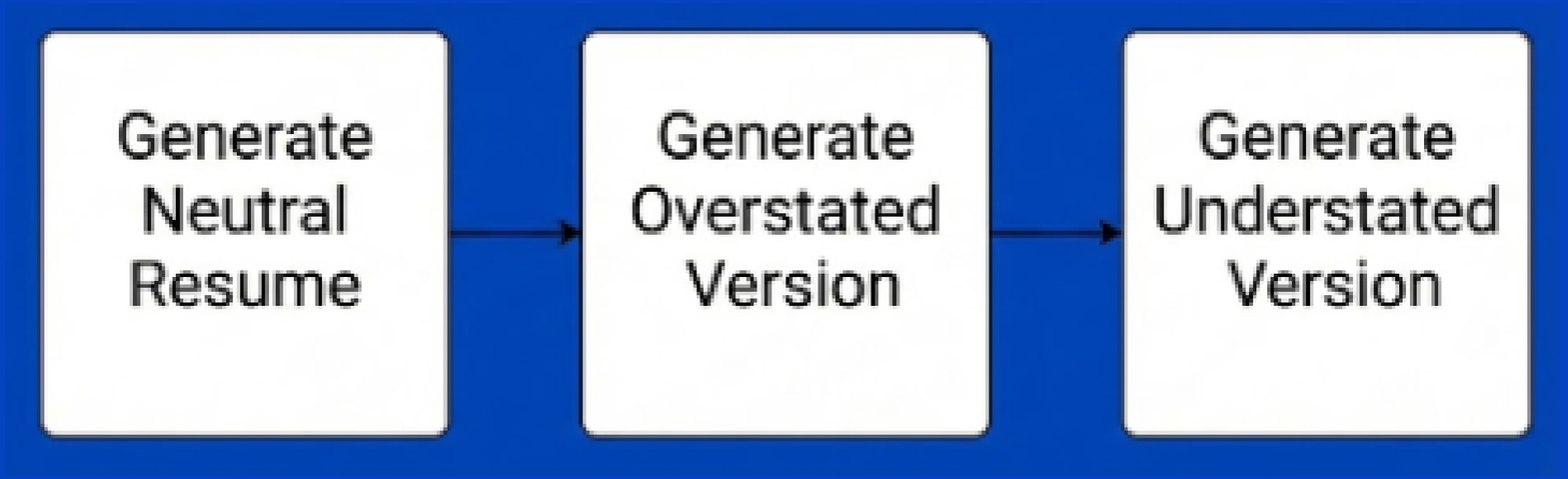
Goal: check stability + fairness, not only accuracy

Test 2 – Style Bias

Data Generation:

- We use Gemma to generate 120 resumes, 40 each class
- Each resume has 3 versions: Neutral, Overstated, Understated
- Total: 120 x 3 = 360 resumes

Process:



Metrics for Test 2 Data

Word Weight Metric

1. Create Power words list and Humble words list (Resources from LinkedIn, Indeed, professional resume writers and other resources)

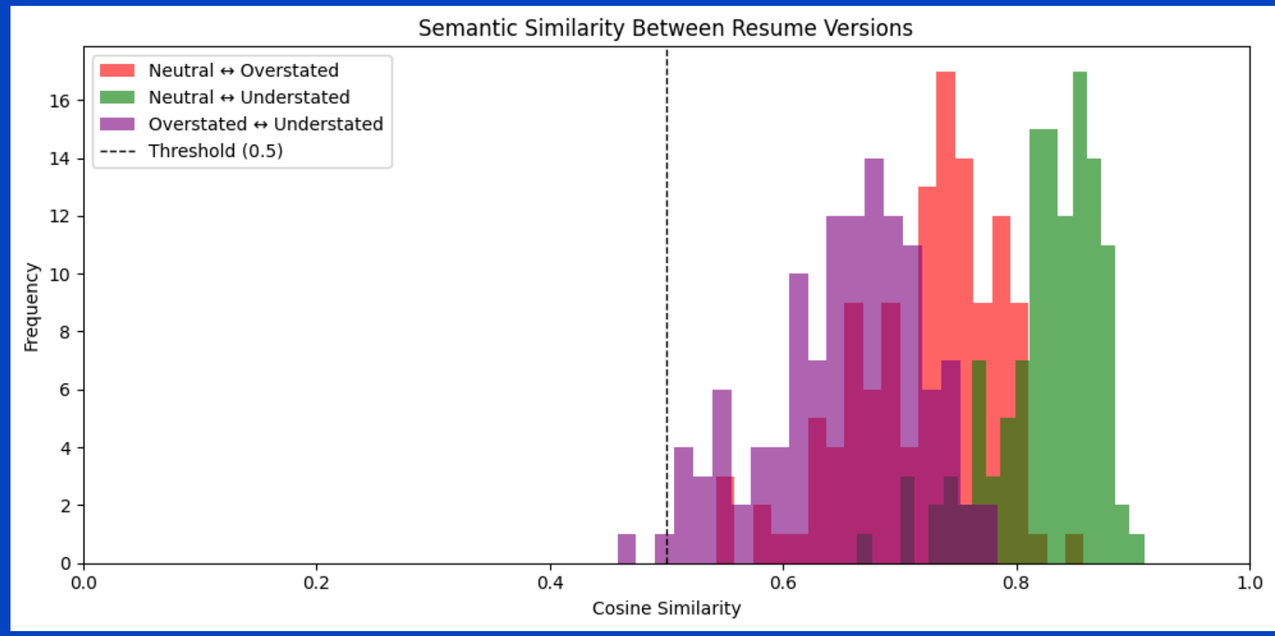
2. Compute the score: $$\text{Tone} = \frac{p - h}{p + h + 1}$$

3. Results:

Style	Tone Score	Avg Power	Avg Humble
overstated	+0.694	23.8	3.9
neutral	-0.383	2.8	6.8
understated	-0.841	0.7	13.1

Semantic Simularity

Model is TF-IDF to convert to vectors



Test 3 – Social Bias (Name Swap Control)

Data Generation:

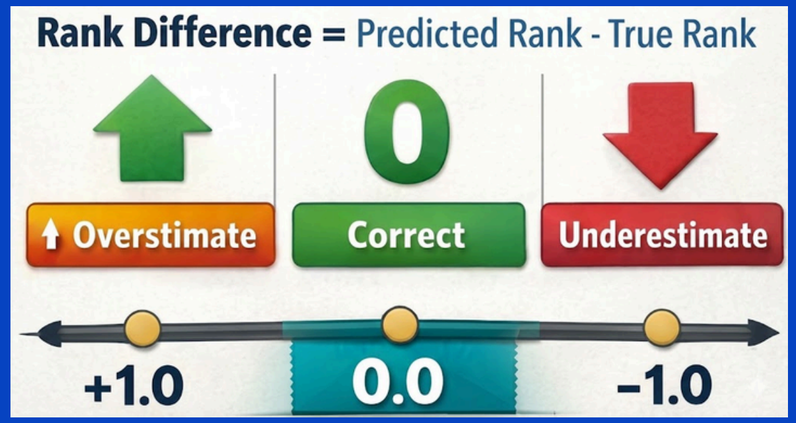
- We use the same neutral resumes from Test 2
- We only change the name using RE.
- Each resume has a name for Caucasian male and female, and a name for African American male and female
- Names from research: Bertrand & Mullainathan (2004) hiring bias study.
- Total: 120 x 4 = 480 resumes

Metrics for Bias & Stability

Accuarcy

How accuare does each model predict resume given to it?

Rank Difference



Does the model systematically overestimare or underestimate seniority?

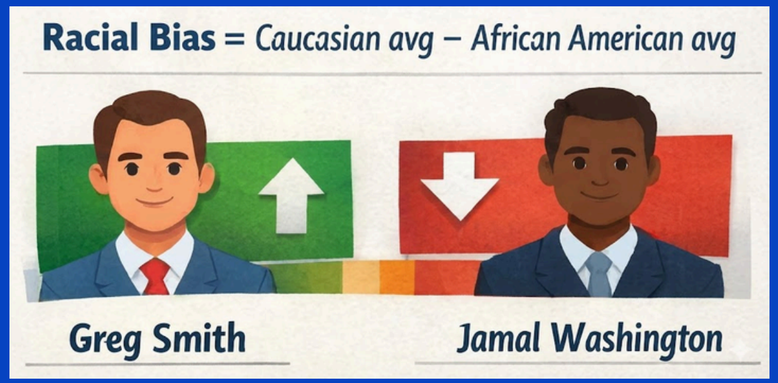
- We Rank Junior, mid and senior as 0,1,2. and copute the score: P - O
- calculate the rank difference for each resume and average at the end.

Inconsistency Rate

Check if the model is inconsistent for each style given to it.

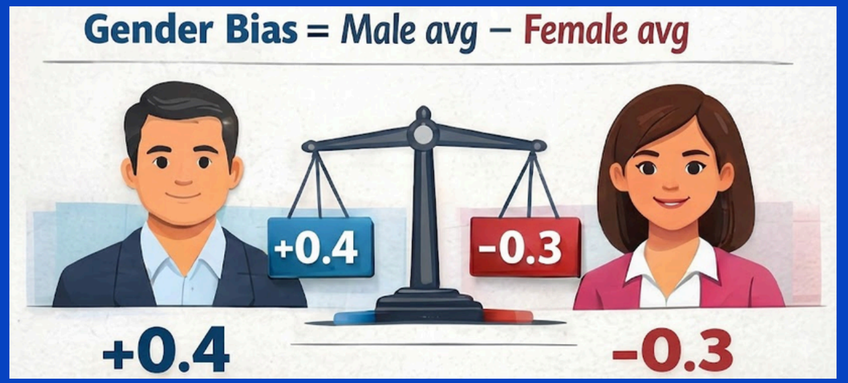
- Count the amount of times the model predicted differently for 2 styles
- Divide by the amount of candidates

Racial Bias Score



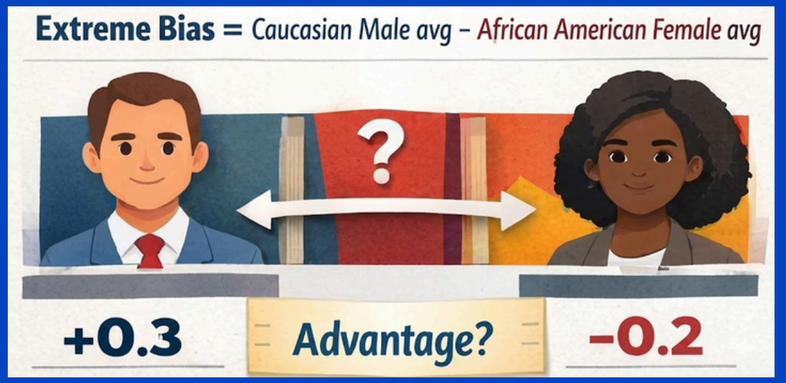
1. Calculate rank difference
2. Group predictions by **race**
3. Calculate avg rank difference
4. Compare the two averages

Gender Bias Score



1. Calculate rank difference
2. Group predictions by **gender**
3. Calculate avg rank difference
4. Compare the two averages

Extreme Bias Score



1. Calculate rank difference
2. Group predictions to **CM and AFF**
3. Calculate avg rank difference
4. Compare the two averages

Test Results

Test 1

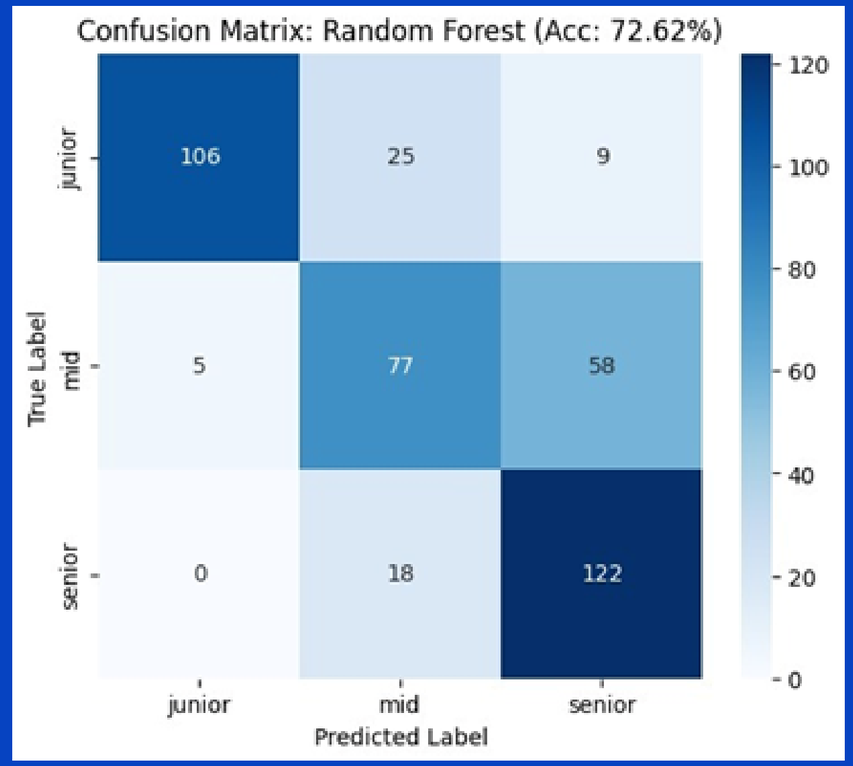
Baseline (TF-IDF + Random Forest)

Metrics:

- Accuracy: 72.6%
- Macro-F1: 0.73
- F1 by class: Junior 0.84 | Mid 0.59 | Senior 0.74

What we found:

- Main confusion is Mid ⇔ Senior.
- Junior is more reliably separable than Mid.
- Baseline is strongly driven by explicit duration signals: removing Years of Experience drops accuracy to 55%.



Test 1

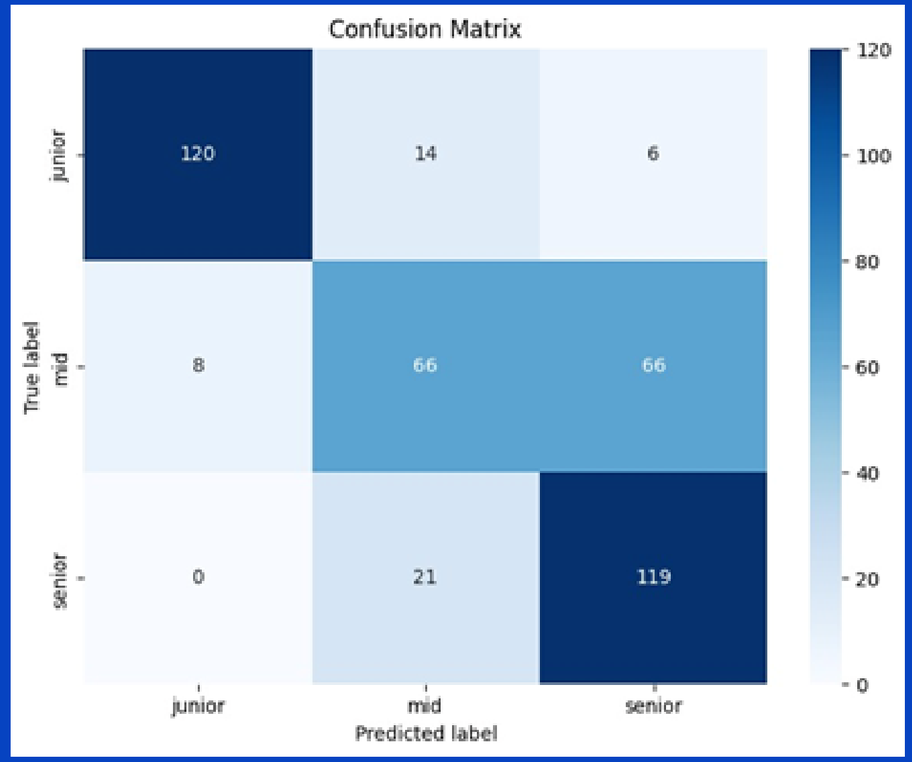
DistillBert

Metrics:

- Accuracy: 72.6%
- Macro-F1: 0.72
- F1 by class: Junior 0.90 | Mid 0.55 | Senior 0.72

What we found:

- Training on single columns gives bad performance (~50%).
- Concatenating all fields into one text input is what makes the model work well.
- Dominant error: Mid → Senior



Test 1

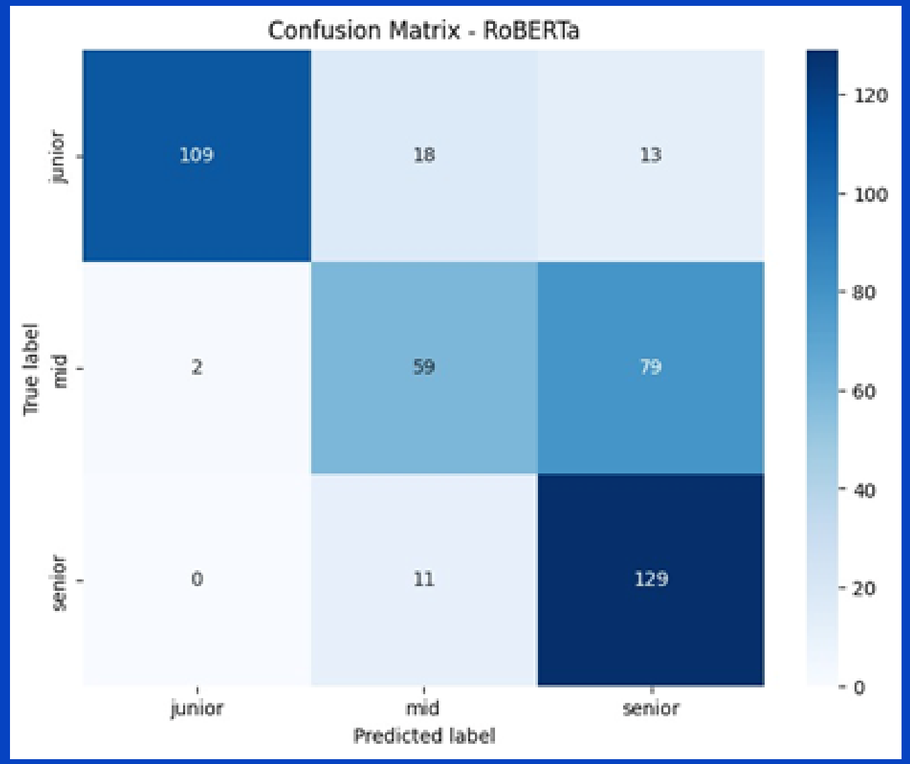
Roberta

Metrics:

- Accuracy: 70.7%
- Macro-F1: 0.70
- F1 by class: Junior 0.87 | Mid 0.52 | Senior 0.71

What we found:

- Same pattern as DistilBERT: single columns gives bad results
- Most errors occur between Mid and Senior.
- Junior and Senior are comparatively stable, with the boundary concentrated in Mid.



Test 1

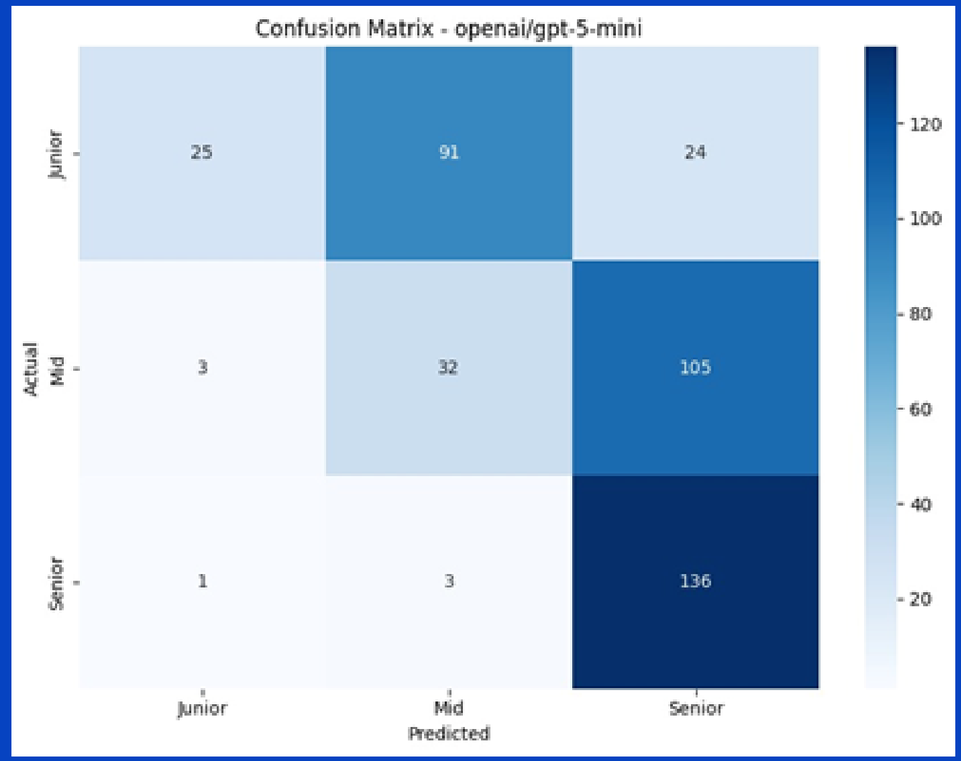
GPT-5

Metrics:

- Accuracy: 45.9%
- Macro-F1: 0.403
- F1 by class: Junior 0.296 | Mid 0.241 | Senior 0.672

What we found:

- Strong overestimation pattern: Junior → Mid, Mid → Senior.
- Senior is relatively strong.
- Suggests the model understands ordinal seniority progression, but uses a high threshold for Junior.



Test 1

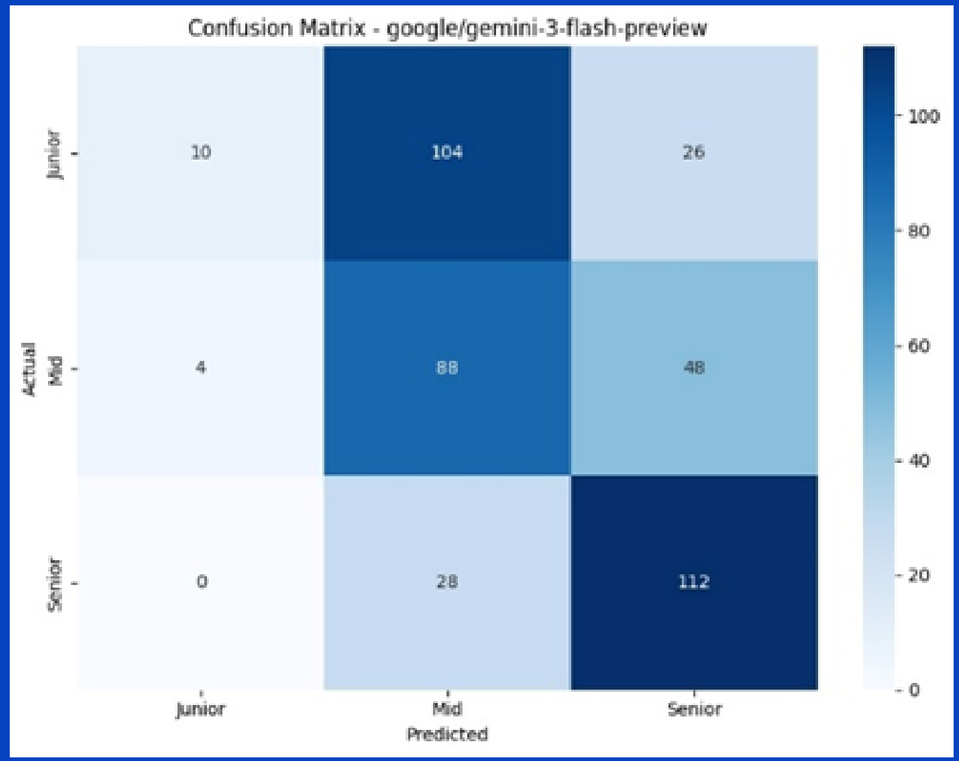
Gemini 3 pro

Metrics:

- Accuracy: 50.0%
- Macro-F1: 0.435
- F1 by class: Junior 0.130 | Mid 0.489 | Senior 0.687

What we found:

- Junior is rarely predicted correctly.
- Errors concentrate in Junior → Mid and Mid → Senior shifts.



Test 1

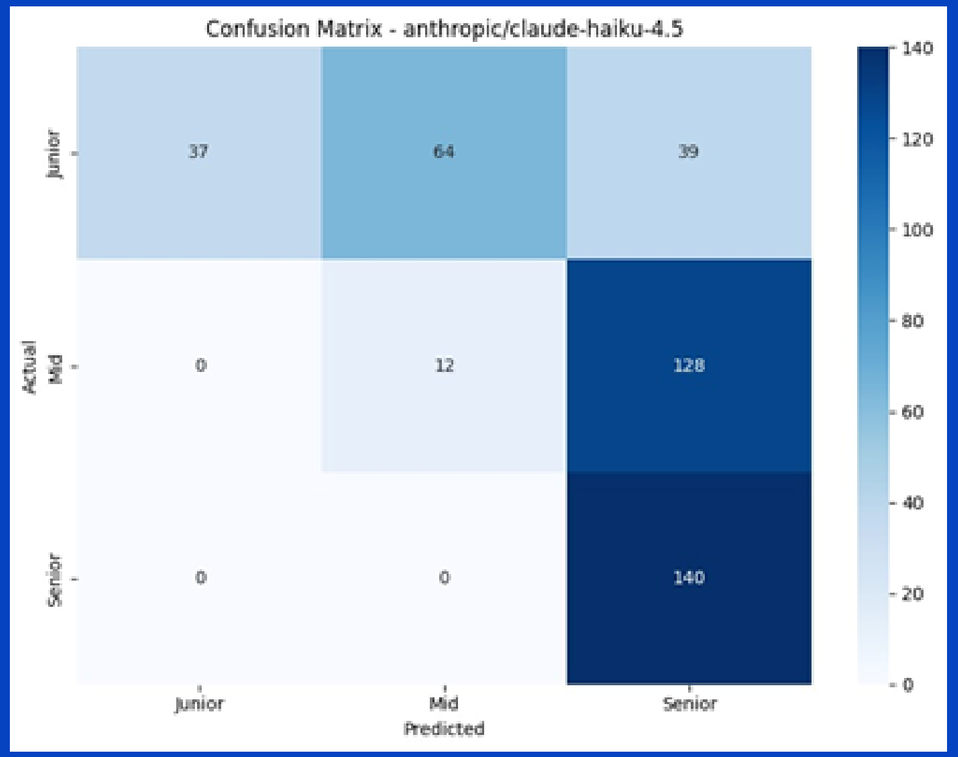
Claude Sonne 4.5

Metrics:

- Accuracy: 45.0%
- Macro-F1: 0.385
- F1 by class: Junior 0.418 | Mid 0.111 | Senior 0.626

What we found:

- Very strong upward bias.
- Senior is predicted perfectly in the confusion matrix, but Mid is extremely weak.



Test 1 Conclusion

Supervised models perform best:

- TF-IDF + Random Forest is the top overall performer (Acc 0.726, Macro-F1 0.73)
- DistilBERT is very close (0.726 / 0.72)
- RoBERTa slightly lower (0.707 / 0.70)

Main difficulty across all supervised models:

- Mid is consistently the hardest class to predict.

Zero-shot LLMs underperform:

- Much lower Macro-F1 overall (~0.39–0.44)
- Gemini 3 Pro is the best among LLMs (Acc 0.50, Macro-F1 0.435)
- GPT-5 + Claude show an upward bias (predicting higher seniority): weak Junior/Mid, relatively stronger Senior

Overall Model Comparison (Test Set)

Model	Accuracy	Macro F1	Junior F1	Mid F1	Senior F1
TF-IDF Baseline	**0.726**	**0.730**	0.730	**0.590**	**0.740**
DistilBERT	**0.726**	**0.730**	**0.90**	**0.590**	**0.740**
RoBERTa	0.707	0.700	0.700	0.520	0.710
GPT-5	0.459	0.402	0.402	0.230	0.670
Claude Sonnet 4.5	0.450	0.385	0.385	0.110	0.630
Gemini 3	0.500	0.435	0.420	0.110	0.690
Gemini 3 pro	0.500	0.435	0.130	0.490	0.690

Test 2

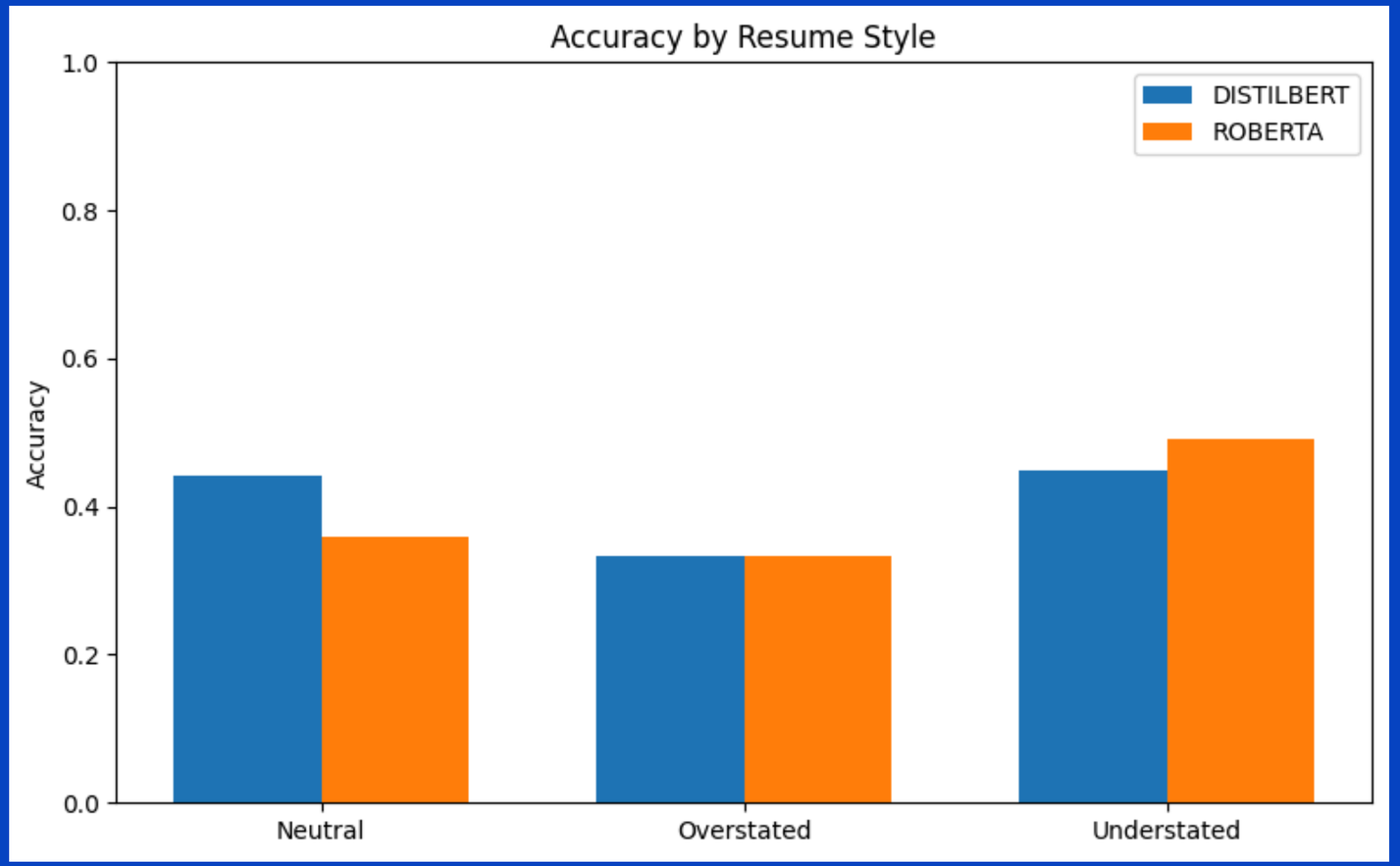
Style Bias

Fine Tuned Models

Accuarcy:

Key findings:

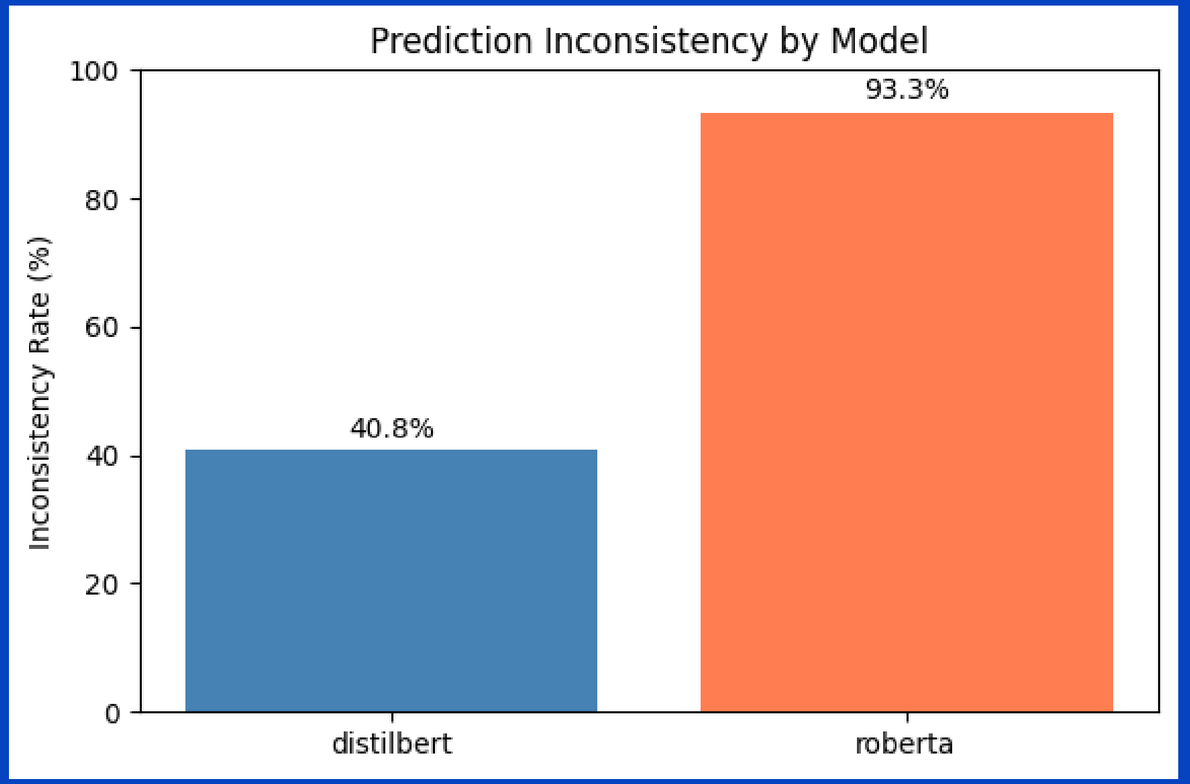
- Overstated drops to 34%. Power words confuse the models.
- Not as high as the dataset, which hints at a pattern in the original dataset.



Inconsistency Rate

Key findings:

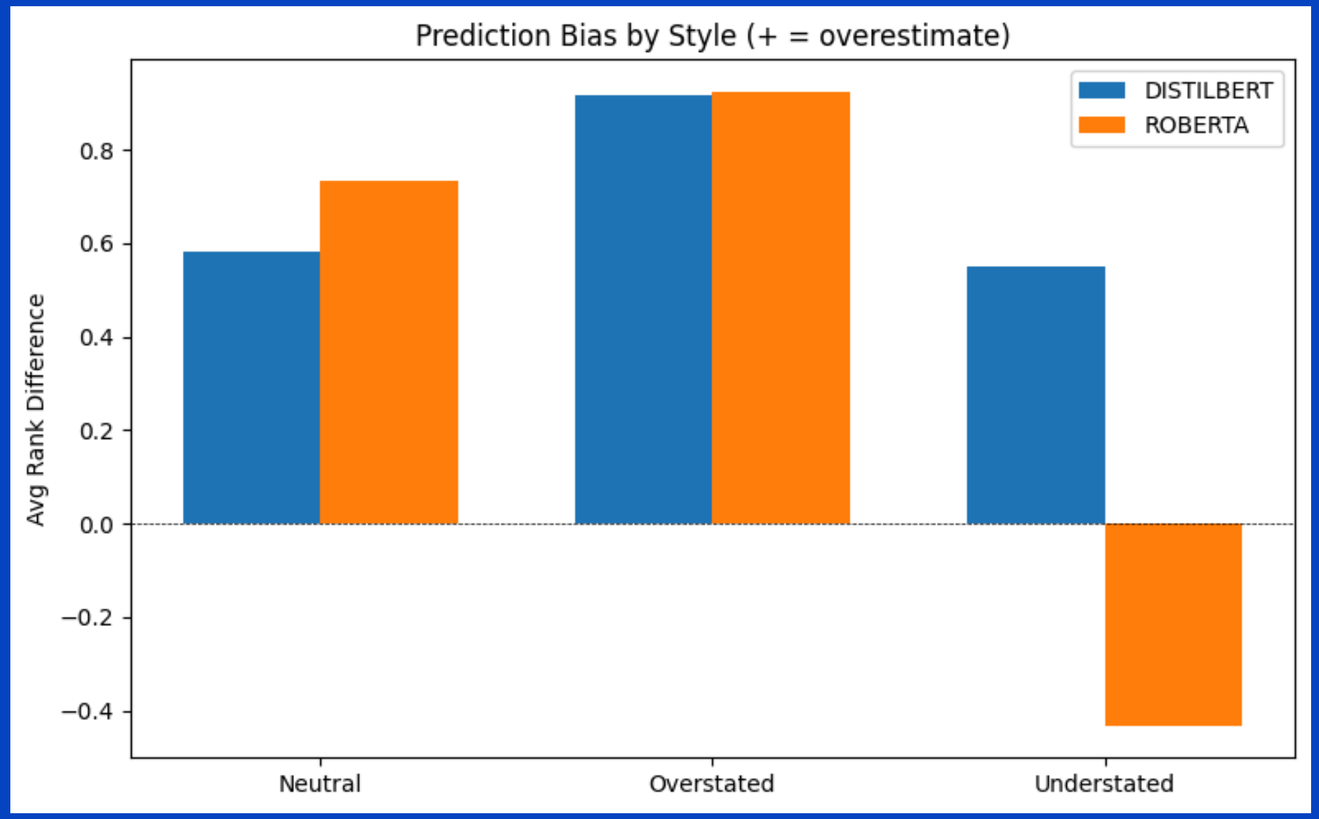
- RoBERTa gives different answers to 93% of people based on style alone.
- Fine-Tuned models are inconsistent when predicting seniority on different styles.



Rank Difference

Key findings:

- Overstated resumes get predicted almost 1 full level too high.
- A junior with an overstated resume looks like a mid or senior for the fine-tuned models.
- The models tend to predict candidates above their true level.
- DistilBERT shows an upward bias even when the resume is understated

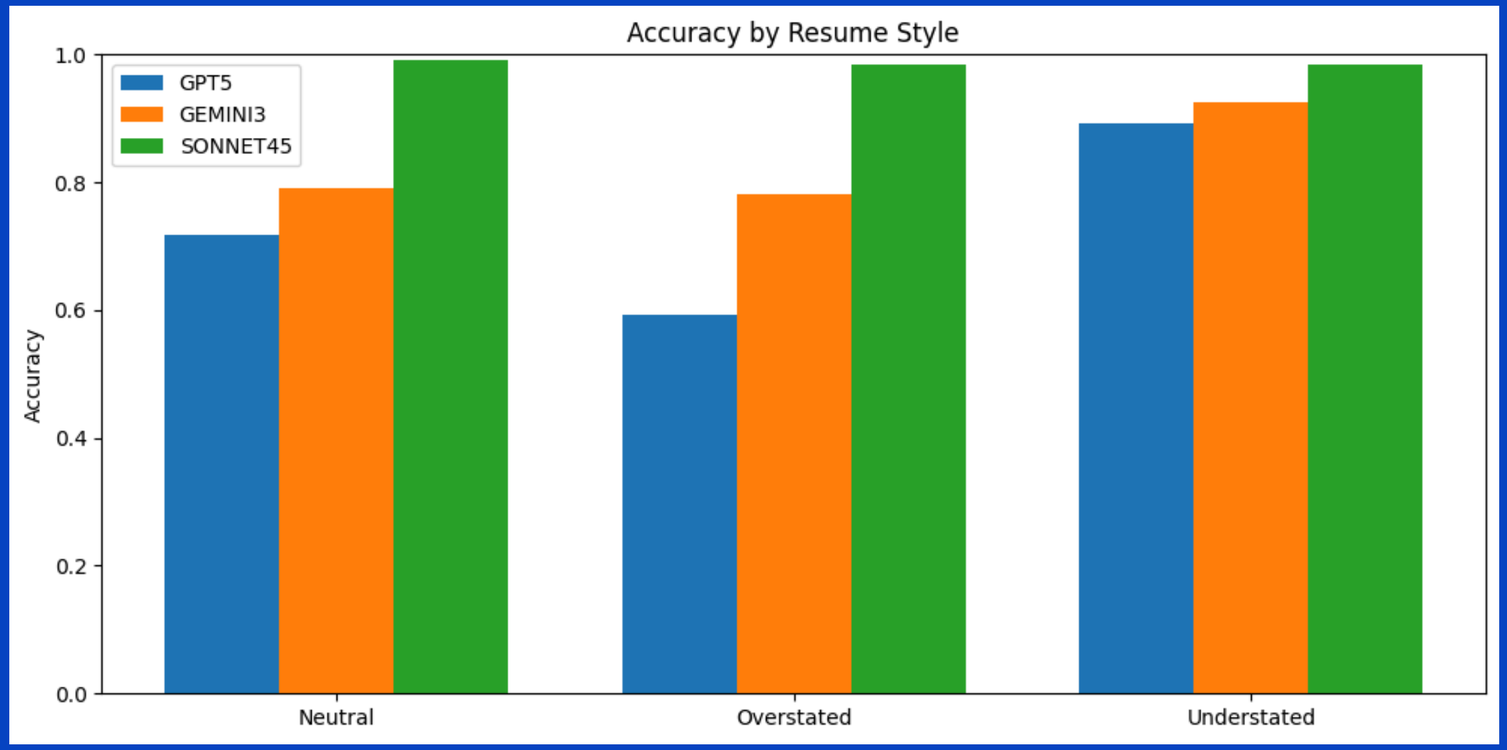


LLMs

Accuracy

Key findings:

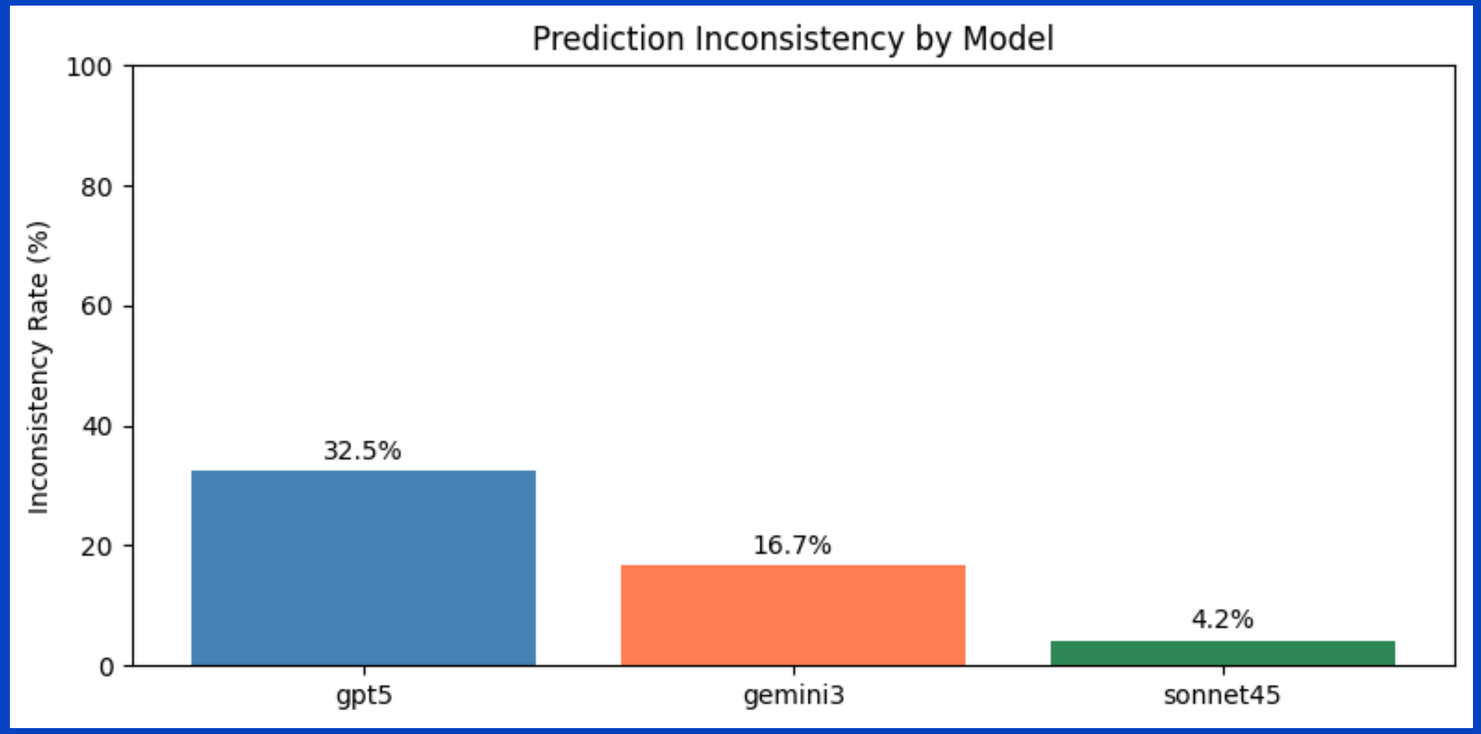
- Sonnet 4.5 Achieve very high accuracy regardless of style.
- Much better accuracy than Test 1, which suggests a hidden pattern that the models have not picked up.
- GPT-5 is the most sensitive to Overstated resumes.
- Understated style improves accuracy for all LLMs



Inconsistency Rate

Key findings:

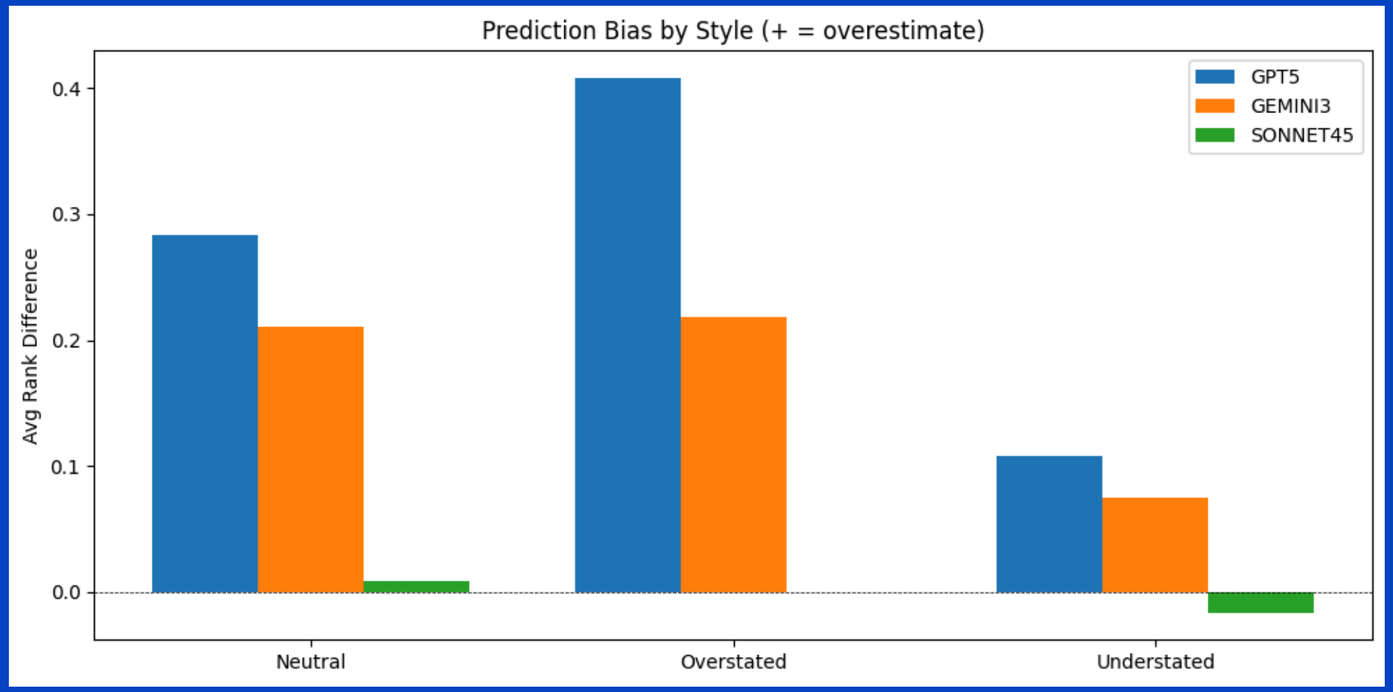
- GPT-5 is the least stable across styles.
- Sonnet 4.5 is highly consistent.
- Matches accuarcy finidings which showed that GPT-5 had the biggest accuarcy swing.
- Gemini 3 was best out of LLMs on test 1, but is not as consistent here, suggesting that it found the hidden pattern in the dataset.



Rank Difference

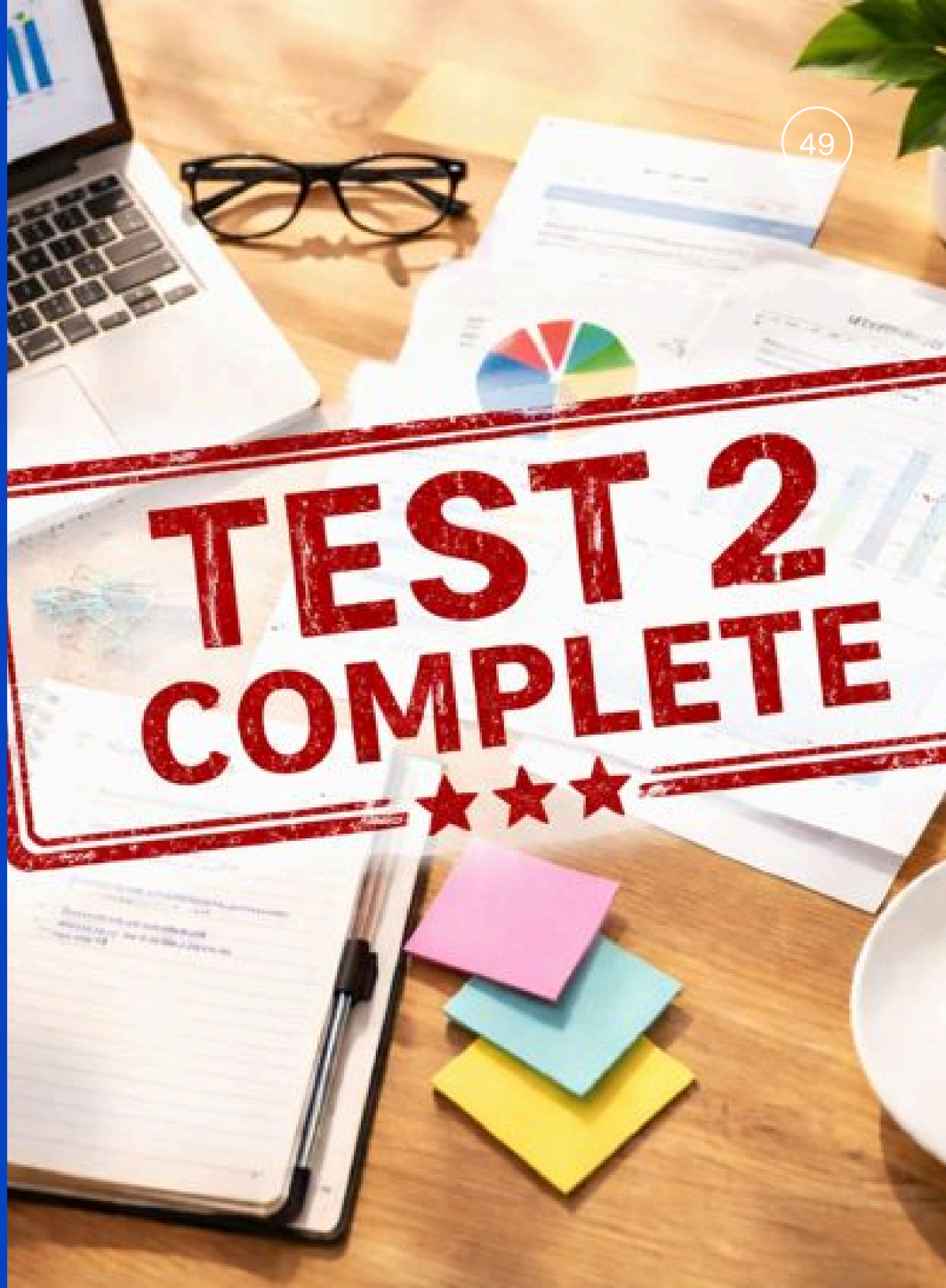
Key findings:

- Overstated style pushes predictions upward for GPT-5 and Gemini 3.
- GPT-5 rewards confident wording in resumes.
- Sonnet 4.5 is essentially unbiased and style-robust.



Test 2 Conclusion

- Style changes can change seniority outcomes
- LLMs are way better than fine-tuned transformers
- Claude model achieves almost zero bias towards style
- Test 1 had a hidden pattern that LLMs could not find
- OpenAI’s GPT ranks lowest out of SOTA models for style bias
- Fine-tuned transformers are pattern fit, not decision ready



Test 3

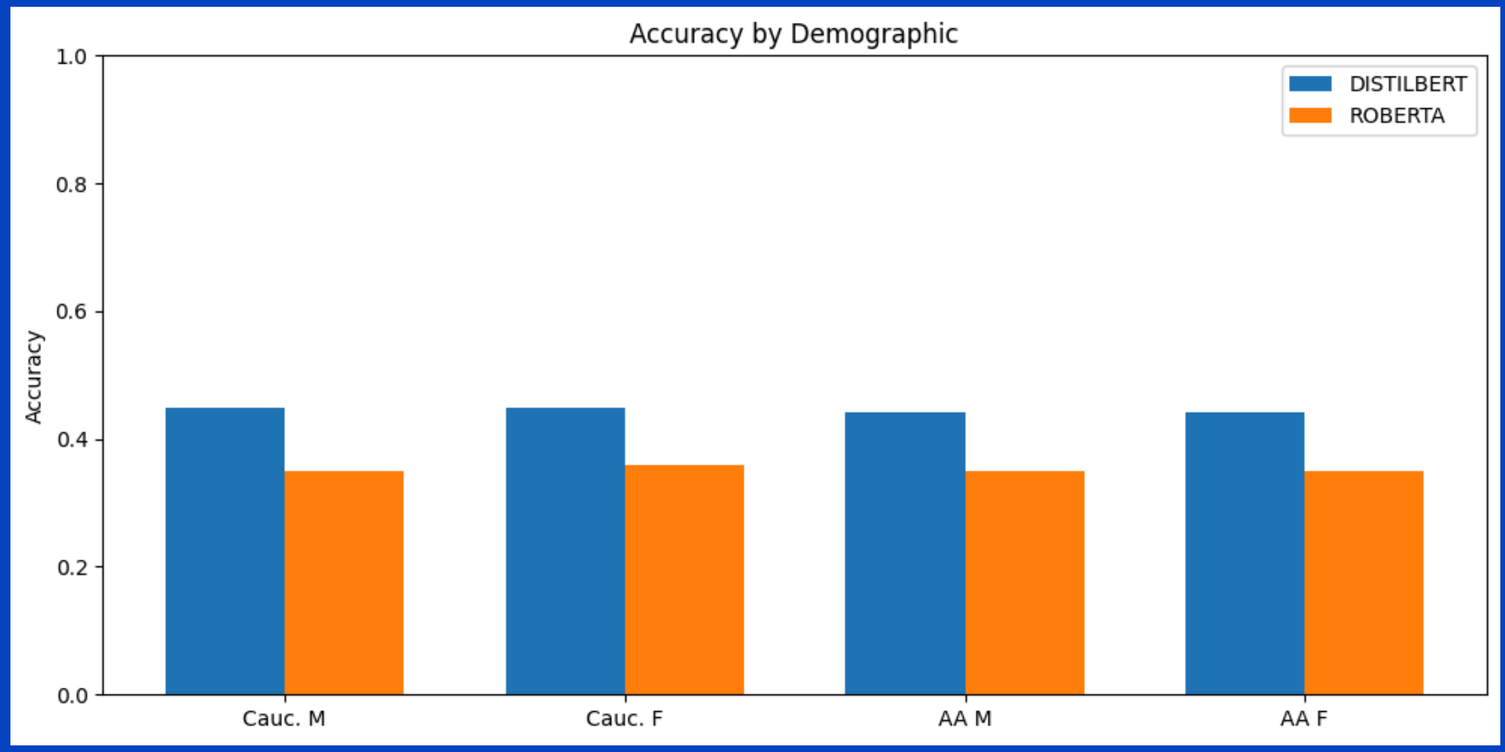
Social Bias

Fine Tuned Models

Accuarcy:

Key findings:

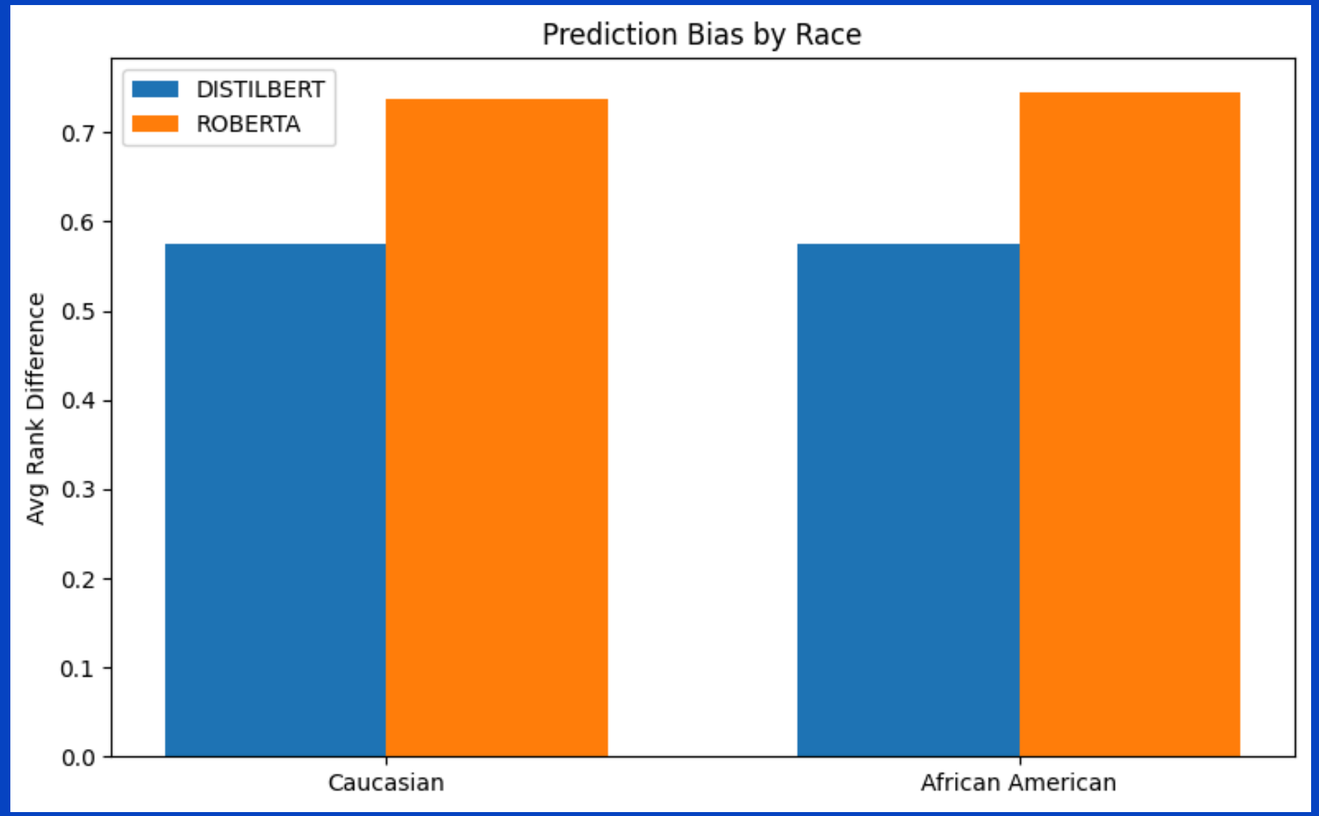
- Accuracy is low overall for both fine-tuned models
- Accuracy is almost identical across demographics
- RoBERTa is consistently worse than DistilBERT across every group



Racial Bias Score

Key findings:

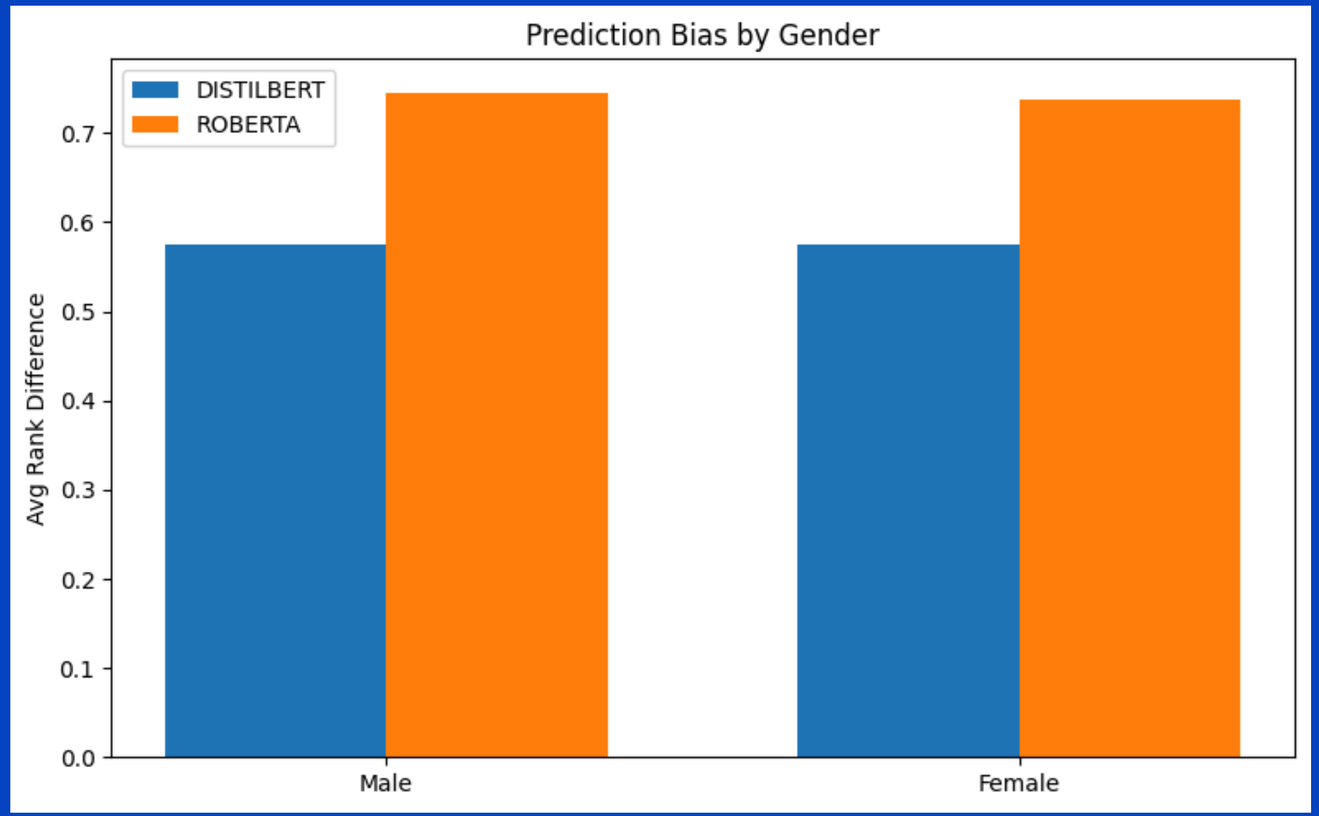
- Both fine-tuned models systematically overestimate seniority for both race groups
- No meaningful race gap in rank difference



Gender Bias Score

Key findings:

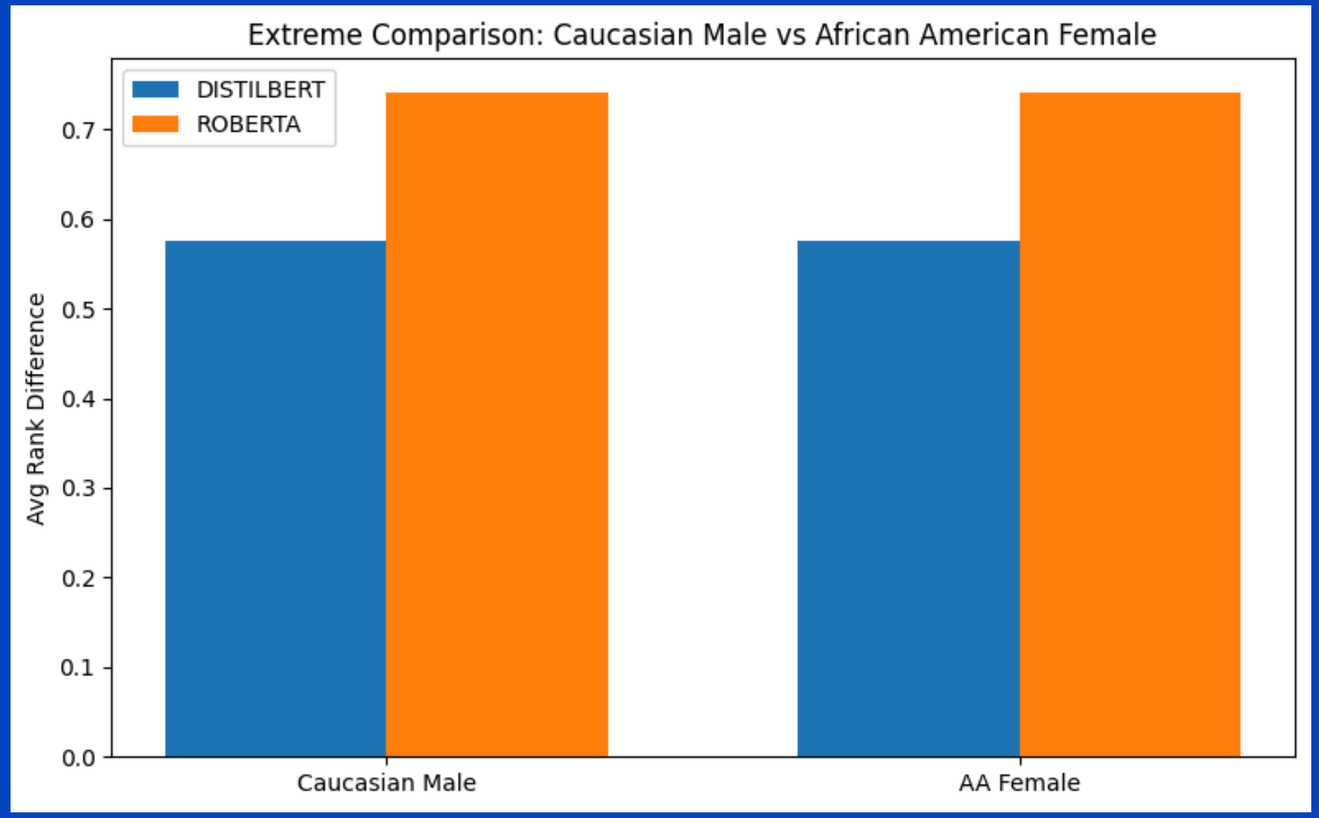
- Strong upward bias for both genders in both models
- Same pattern across race and gender. Failure is global, not demographic-specific



Extreme Bias Score

Key findings:

- No visible extreme-group gap in rank difference
- The same upward bias problem dominates again

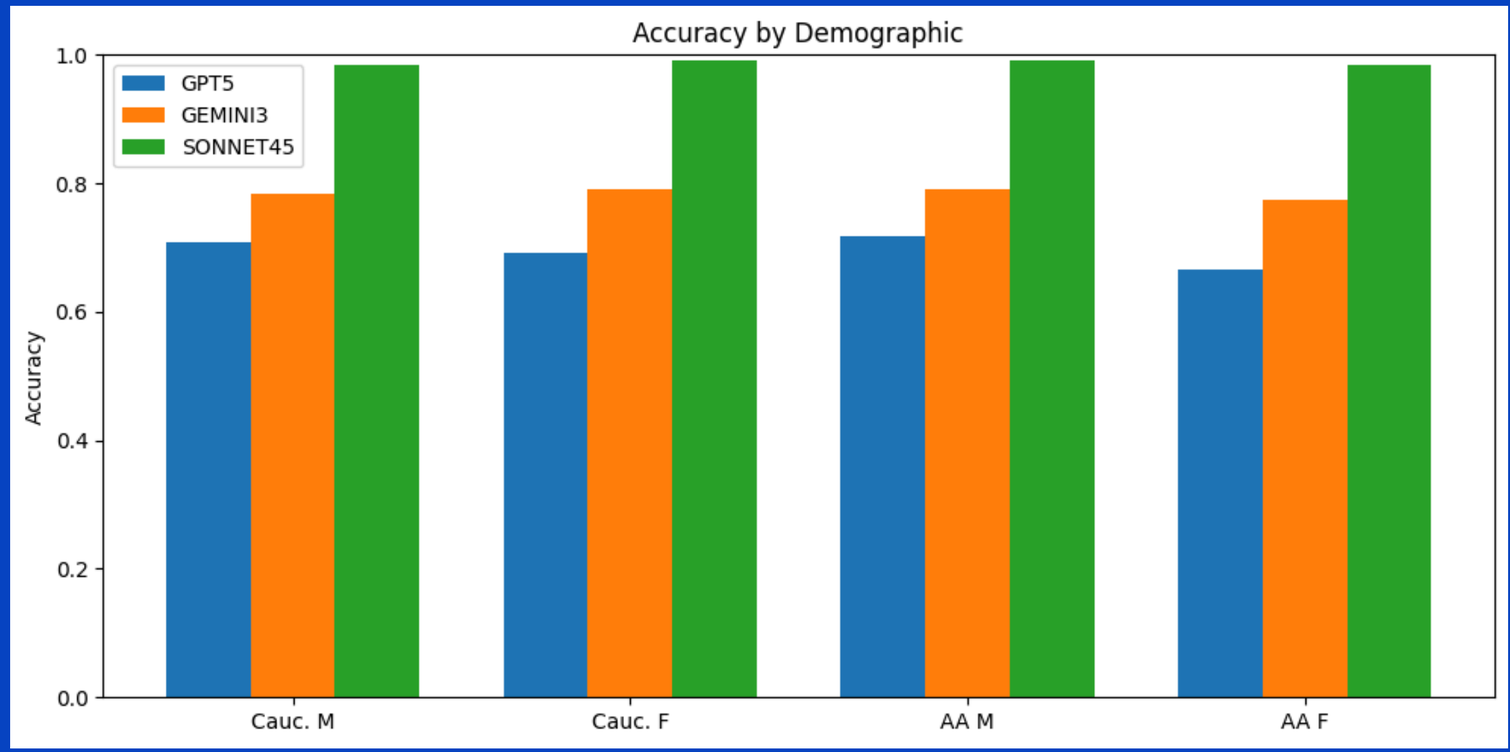


LLMs

Accuracy

Key findings:

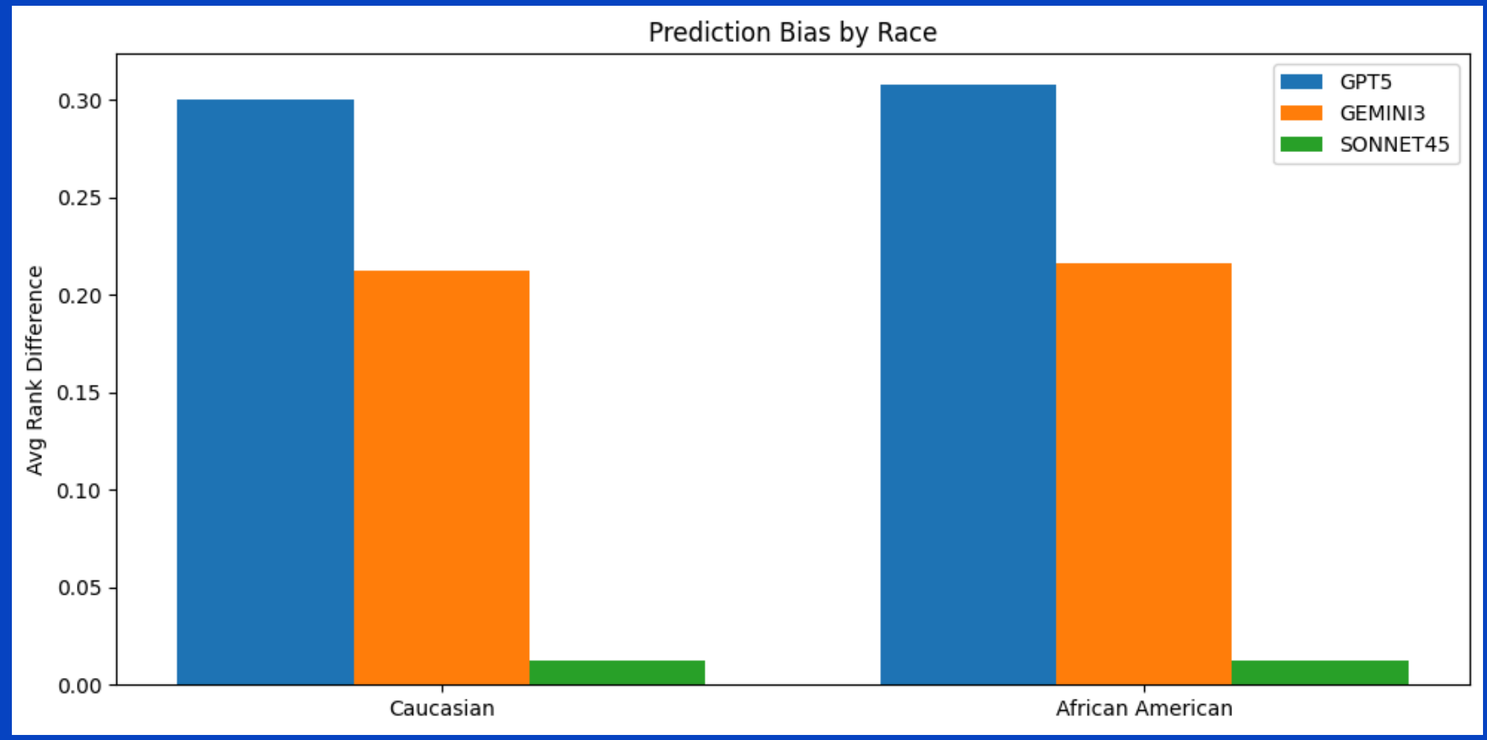
- Sonnet 4.5 is near-perfect and consistent across all demographics
- All models are consistent, suggesting zero bias within the setup



Racial Bias Score

Key findings:

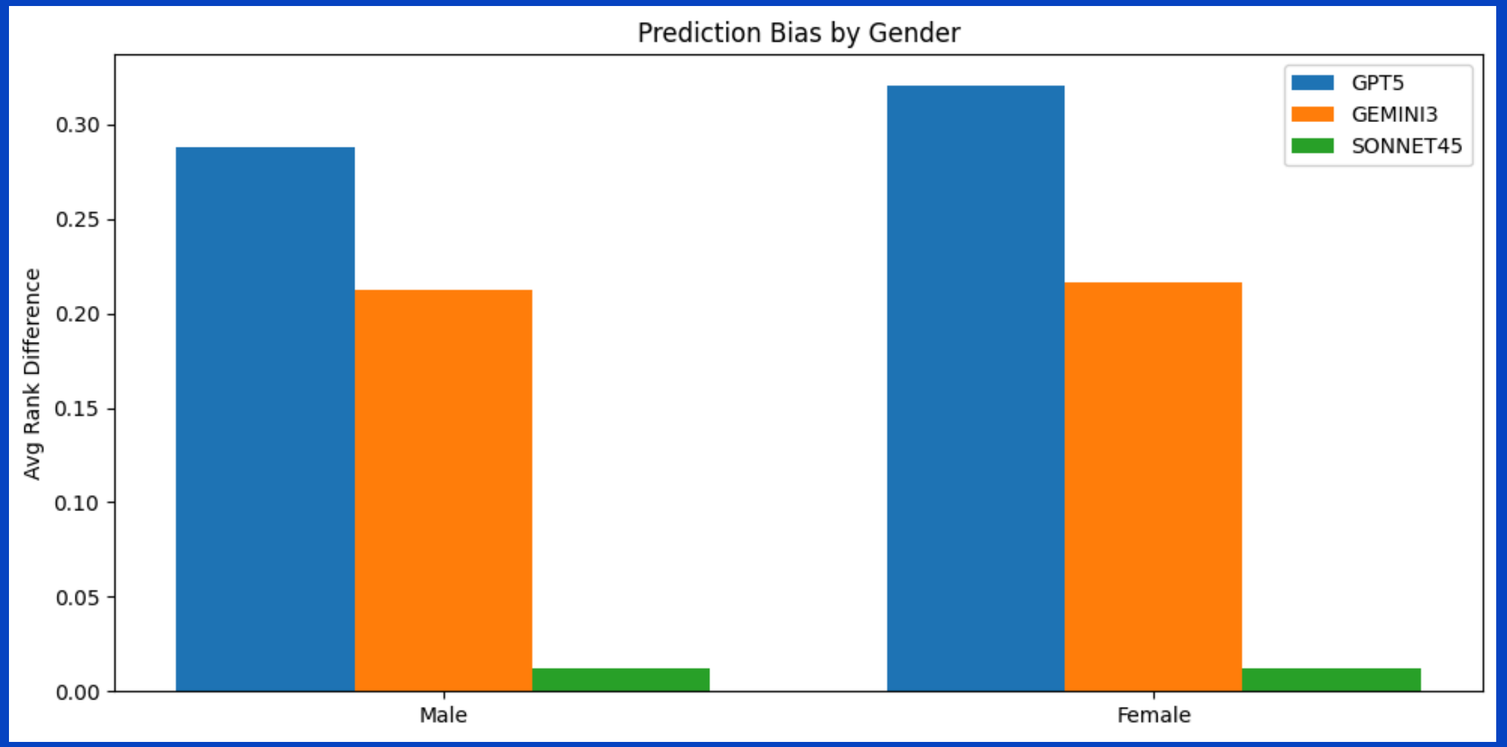
- No meaningful race gap for any LLM
- GPT-5 and Gemini 3 still show an overall upward bias
- Sonnet 4.5 is closest to being unbiased



Gender Bias Score

Key findings:

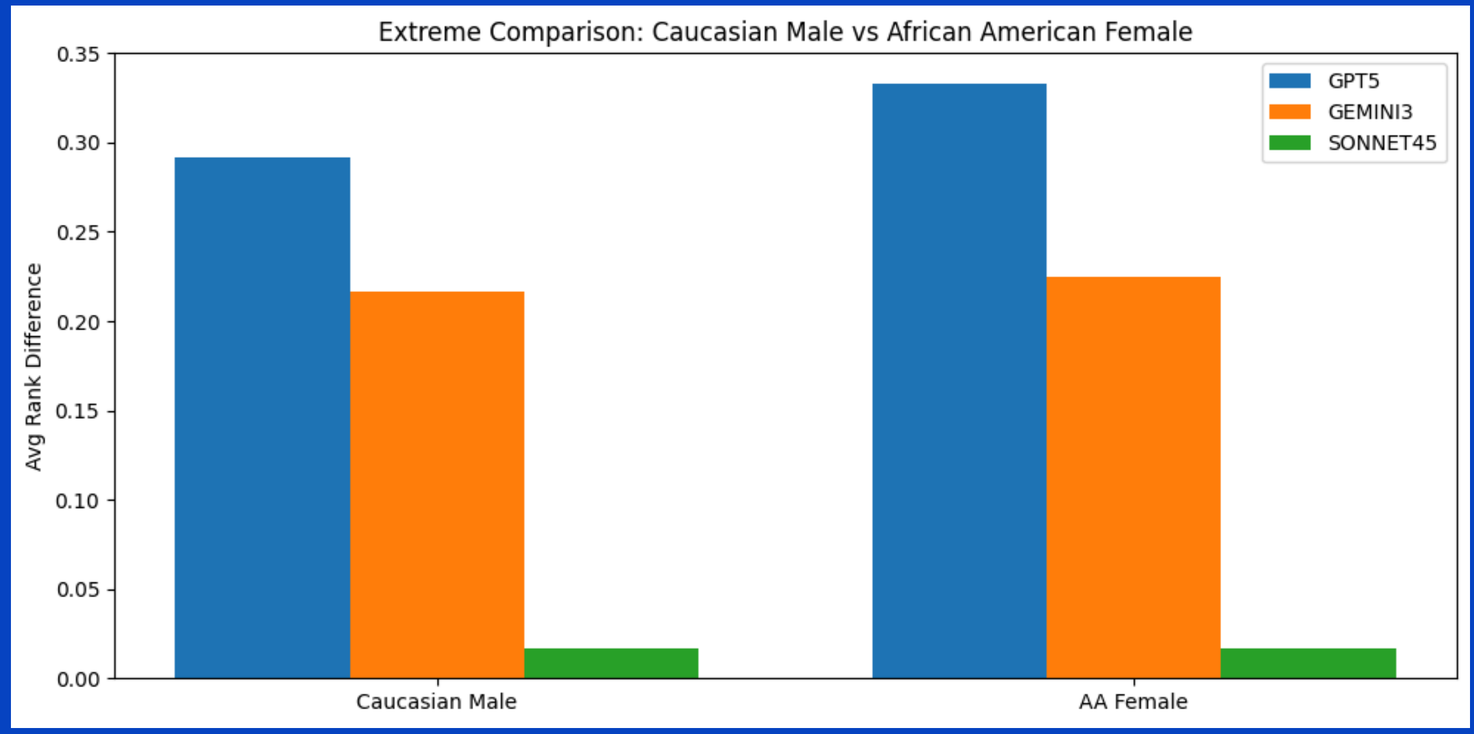
- No meaningful gender gap for Gemini and Sonnet
- GPT shows a small gender-coded difference, but the dominant effect is still upward bias overall



Extreme Bias Score

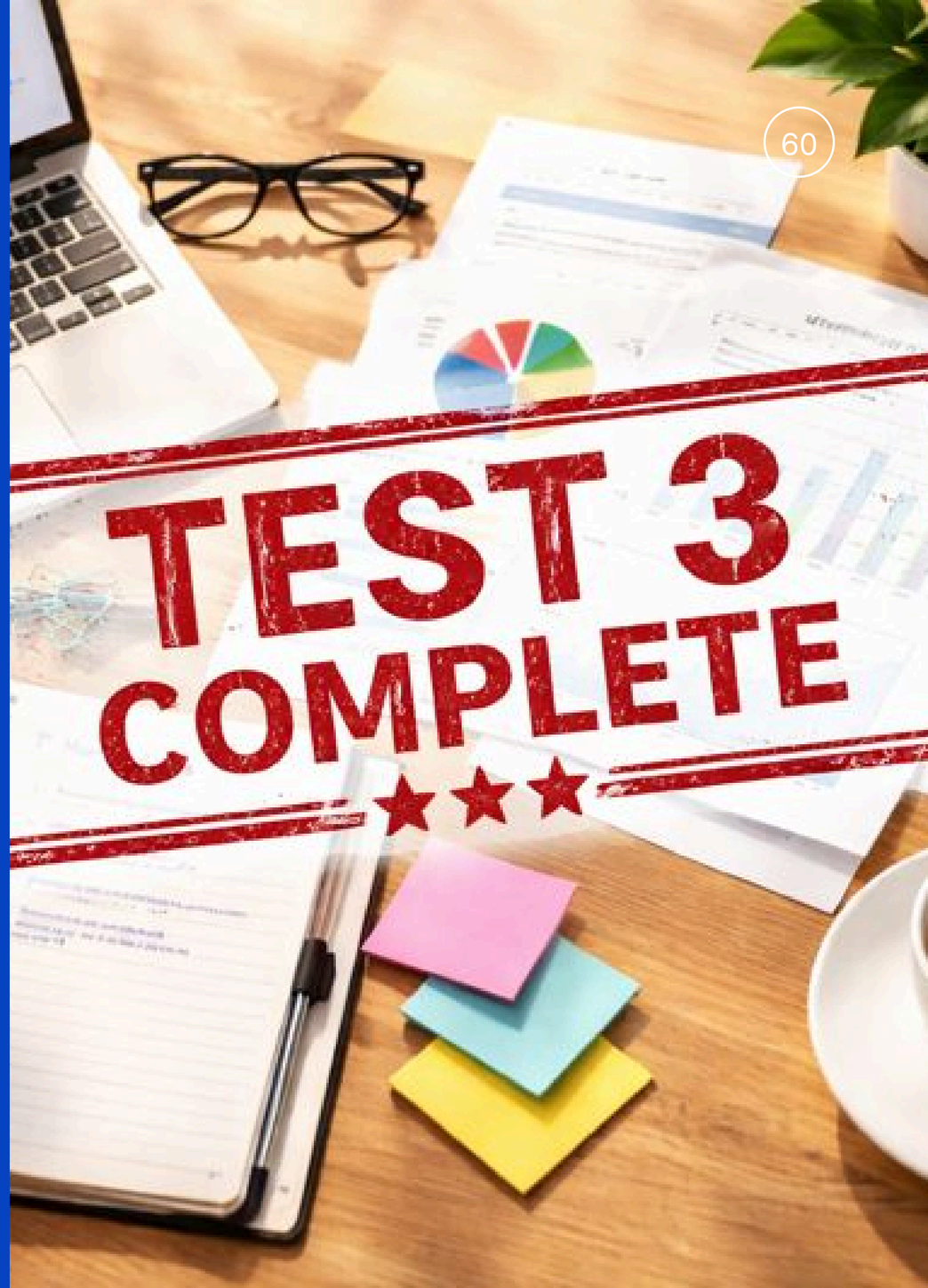
Key findings:

- No intersectional disadvantage appears in this test
- GPT-5 slightly favor to African American females
- Sonnet 4.5 stays near zero for both extremes



Test 3 Conclusion

- LLMs are more accurate than fine-tuned transformers
- No clear race based bias was observed in rank shift metrics, with GPT showing a small shift
- Test 1 had a hidden pattern that LLMs could not find
- Very good improvements from older LLM models that suggested social biases
- Fine-tuned transformers are not deployment-ready here
- Fine-tuned models seem to rely on surface patterns from the training data, so when the input is rewritten in tone or style, reliability drops



Limitations of the Study

DATASET CHOICE

Zero-shot LLM underperformance in this test 1 setup. Prompt-only evaluation of foundation LLMs (GPT-5, Claude Sonnet 4.5, Gemini 3 Pro) did not reach the performance of supervised approaches, limiting their immediate applicability as drop-in classifiers for this dataset.

MID CLASS

Mid-level ambiguity and label separability. Across all model families, Mid-level candidates were the hardest to classify, with confusion matrices showing frequent overlap with both Junior and Senior.

RESOURCE CONSTRAINTS

Budget and hardware limitations restricted the experimental scope. We could not fine-tune larger foundation models, run extensive hyperparameter searches, or scale up prompt-based experiments (e.g., few-shot variants) due to API costs and compute availability.

Future work

NOTES FOR FUTURE RESEARCH

Prompting extensions for LLMs. Evaluate few-shot prompting and structured prompting variants (while keeping a consistent protocol) to test whether calibration and decision boundaries improve relative to strict zero-shot.

Model and representation improvements. Explore additional supervised variants such as domain-adaptive pretraining, alternative transformer backbones, and hybrid models that combine structured numeric signals with contextual text representations under explicit leakage control

Deeper error analysis for Mid-level cases. Perform targeted qualitative and quantitative analysis of Mid-level false positives/negatives to identify which textual signals drive confusion (e.g., leadership verbs, scope language, project ownership, team size, impact metrics), and use these insights to refine feature construction and evaluation.

Calibration and thresholding. Apply simple calibration approaches (e.g., class-threshold adjustments or post-hoc calibration on validation data) to reduce consistent upward bias observed in zero-shot LLM predictions and improve Mid-level discrimination

Thank you for listening!