Algorithmic Bias in AI Resume Screening

AUTHOR
Aidan Bugayong

AFFILIATION
Department of Mathematics, Applied Mathematics, and Statistics, Case Western Reserve University

PUBLISHED
November 14, 2025

1 Introduction

1.1 Context and Background

Resume screeners were developed in order to screen canidates more efficiently and reduce the human bias in the screening process. However, there are concerns with whether or not these automated resume screening systems are truly unbiased in their decision making process. As more and more companies use some form of AI automation in their hiring process, the question of whether or not these systems are unbiased has become more important.

According to Naveen Kumar in his article on AI recruitment statistics, roughly 87% of companies are using AI in the hiring and recruitment process[1]. As such, the University of Washington decided to conduct their own study to determine if there is any bias in the decision making process of resume screeners and what sorts of factors contribute to the scoring process of a resume. Their research found “significant racial, gender and intersectional bias in how three state-of-the-art large language models, or LLMs, ranked resumes. The researchers varied names associated with white and Black men and women across over 550 real-world resumes and found the LLMs favored white-associated names 85% of the time, female-associated names only 11% of the time, and never favored Black male-associated names over white male-associated names”[2]. Their research came to these conclusions by handing the AI a list of identical resumes with different names and then having the AI give them scores. The article ends with the research team noting that more research should be done in this area by looking at different attributes and more LLMs in order to better allign these AI systems with the real world policies to reduce bias and harm.

1.2 Project Purpose

This project aims to determine if there is any bias in the decision making process of resume screeners and what sorts of factors contribute to the scoring process of a resume. More specifically, looking at a wide variety of applicant attributes to determine what factors have the highest contribution to the bias in the decision making process. Ideally, professional experience and education will be the best predictors of the resume scores but we will also look into other factors such as gender, ethnicity, and institutional prestige.

2 Methods

2.1 Data Collection and Purpose

The original list of resumes is from a dataset in huggingface[3], which is comprised of both real and synthetic resume data in JSON formatting. The purpose of this dataset was for training natural language processing (NLP) models for resume parsing. Specifically, the resumes are oriented around technical roles and is designed for NLP models to be trained on and used for candidate matching / screening in this field. This dataset was posted on huggingface Feburary 21st, 2025 and has not been updated since (excluding the minor changes to the readme).

The sources used for this dataset are sourced from anonymized CV submissions as well as synthetic resumes generated using “Faker Library” and filled with realistic and role appropriate information. All resumes are anonymized by removing PII (Personally Identifiable Information) but many fields (such as names) contain realistic placeholders. The makers of the dataset note that the data is oriented around technical roles and the synthetic resumes may not capture the same nuances of a real resume. As such, the makers note that this dataset should only be used for NLP, data augmentation, or exploratory data analysis and should not be used for non-technical roles or personalized hiring decisions.

The dataset contains over 4500 resumes in a JSON format. Each resume entry contains personal information, work experience, education information, skills and projects. Since these are technical resumes (oriented around the computer science / information technology field), the skills and projects fields contained a candidate’s coding projects and/or coding languages. For the scope of this project, all fields were used for analysis or scoring of the resume.

2.2 Data Processing

The collection of resumes was processed in order to create the score datasets used for this project. The score datasets has the following columns: Name (acting as the primary key), Resume Score, Gender, Ethnicity, Institutional Prestige, Years of Experience, Skill Relevance, Experience Relevance and Project Relevance. The score datasets are specific to each of the langauge models used in this project. This is because we are trying to understand the relation between how the model scores a reseume and the model’s own perception of the candidate (gender, ethnicity, and institutional prestige).

For creating the score datasets themselves, the resumes were scored by the model and then the model was asked to guess the gender, ethnicity, and institutional prestige of the candidate. For determining the skill/project/expperience relevance, that was also a call to the language model. It is important to note that each column of data is a separate instance of the language model to reduce the liklihood of previous responses affecting the current responses. All of this information was then used to create the score dataset for each of the models used in this project.

The models used for this project include IBM’s *granite3.3:8b*, Microsoft’s *Phi4:14b* and Meta’s *llama3.1:8b*. These models were choosen because they are highly rated models despite the lower paramter size and are all open source.

2.3 Statistical Tools and Approach

After the resumes have been processed by the language models, there was some data cleaning to ensure the models followed the instructions. The initial approach was to use linear regression to predict the resume score from all of the other variables in the dataset. The formula used for this model was: `score ~ gender + ethnicity + prestige + skill_score + project_score + experience_score + years_experience`. Here, the response variable is the resume score which are integers in the range 0-100. The categorical variables are: gender, ethnicity, and prestige. The numeric variables are: years of experience, skill score, project score, and experience score. All categorical variables were setup such that “Unknown” was the reference category. The numeric variables, like the resume score, are in range 0-100, with the exception of years experience which is a positive decimal.

Next there was analysis of interation effects and determining if any interactions were considered statistically significant. If the interaction was considered significant, then the interaction was added to the model. Additionally, the model diagnostics were conducted to ensure that the model's assumptions were met. Specifically, the model was checked for normality in the residuals and multicollinearity by using the Shapiro-Wilk test and variance inflation factor, respectively. Interaction effects were added to the model if they were statistically significant. Due to multicolinearity issues, the model used a ridge regression.

3 Results and Discussion

Initial Modeling

```r
ibm = read.csv("data/granite3.3_8b.csv")
ibm$gender = relevel(factor(ibm$gender), ref = "Unknown")
ibm$ethnicity = relevel(factor(ibm$ethnicity), ref = "Unknown")
ibm$prestige = relevel(factor(ibm$prestige), ref = "Unknown")
meta = read.csv("data/llama3.1_8b.csv")
meta$gender = relevel(factor(meta$gender), ref = "Unknown")
meta$ethnicity = relevel(factor(meta$ethnicity), ref = "Unknown")
meta$prestige = relevel(factor(meta$prestige), ref = "Unknown")
microsoft = read.csv("data/phi4_14b.csv")
microsoft$gender = relevel(factor(microsoft$gender), ref = "Unknown")
microsoft$ethnicity = relevel(factor(microsoft$ethnicity), ref = "Unknown")
microsoft$prestige = relevel(factor(microsoft$prestige), ref = "Unknown")

models=NA

models$ibm = lm(score ~ gender + ethnicity + prestige + skill_score + project_score + experienc
summary(models$ibm)
```

```
Call:
lm(formula = score ~ gender + ethnicity + prestige + skill_score +
    project_score + experience_score + years_experience, data = ibm)

Residuals:
    Min      1Q  Median      3Q     Max
-69.945 -11.519   5.942  13.078  33.777

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)
(Intercept)               77.69718    6.48737  11.977  < 2e-16 ***
genderFemale              -2.25304    2.92340  -0.771   0.4411
genderMale                -6.90248    2.90991  -2.372   0.0179 *
ethnicityAfrican American  8.54478    5.71503   1.495   0.1353
ethnicityAsian             6.52131    6.67507   0.977   0.3289
```

```
ethnicityCaucasian        11.31109    5.45623   2.073   0.0385 *
ethnicityHispanic         10.60837    5.61769   1.888   0.0593 .
prestigeHigh              -3.68607    2.14469  -1.719   0.0860 .
prestigeLow               -3.46246    1.69783  -2.039   0.0417 *
prestigeMedium             0.68682    1.78618   0.385   0.7007
skill_score               -0.10363    0.04589  -2.258   0.0242 *
project_score             -0.07630    0.04012  -1.902   0.0575 .
experience_score          -0.08040    0.03661  -2.196   0.0284 *
years_experience          -0.74379    0.18186  -4.090 4.73e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 18.65 on 838 degrees of freedom
Multiple R-squared:  0.07673,	Adjusted R-squared:  0.06241
F-statistic: 5.357 on 13 and 838 DF,  p-value: 2.473e-09
```

```r
models$meta = lm(score ~ gender + ethnicity + prestige + skill_score + project_score + experier
summary(models$meta)
```

```
Call:
lm(formula = score ~ gender + ethnicity + prestige + skill_score + 
    project_score + experience_score + years_experience, data = meta)

Residuals:
     Min       1Q   Median       3Q      Max 
-22.5771  -3.1530   0.6273   4.9550  11.1384 

Coefficients:
                            Estimate Std. Error t value Pr(>|t|)    
(Intercept)               93.8050893  4.9344847  19.010   <2e-16 ***
genderFemale              -6.3194893  4.6807942  -1.350    0.177    
genderMale                -6.0771735  4.6805950  -1.298    0.194    
ethnicityAfrican American -0.9453498  1.0218348  -0.925    0.355    
ethnicityAsian            -0.3434008  1.2523898  -0.274    0.784    
ethnicityCaucasian        -0.8673318  0.9603392  -0.903    0.367    
ethnicityHispanic          0.3980222  0.9966852   0.399    0.690    
prestigeHigh              -0.4144583  0.6497467  -0.638    0.524    
prestigeLow               -0.0208554  0.5156508  -0.040    0.968    
prestigeMedium             0.1902366  0.5407470   0.352    0.725    
skill_score                0.0007099  0.0084802   0.084    0.933    
project_score              0.0080575  0.0085231   0.945    0.345    
experience_score          -0.0008999  0.0105291  -0.085    0.932    
years_experience           0.0167624  0.0557211   0.301    0.764    
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 6.555 on 1042 degrees of freedom
Multiple R-squared:  0.01017,	Adjusted R-squared:  -0.00218 
F-statistic: 0.8235 on 13 and 1042 DF,  p-value: 0.6354
```

```
F-statistic: 0.8255 on 13 and 1042 DF,  p-value: 0.6354
```

```
models$microsoft = lm(score ~ gender + ethnicity + prestige + skill_score + project_score + exp
summary(models$microsoft)
```

```
Call:
lm(formula = score ~ gender + ethnicity + prestige + skill_score + 
    project_score + experience_score + years_experience, data = microsoft)

Residuals:
    Min      1Q  Median      3Q     Max 
-40.737  -4.729   1.004   6.369  23.726 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)               32.87479    4.89587   6.715 3.07e-11 ***
genderFemale              -2.40881    2.71757  -0.886  0.37561    
genderMale                -2.07788    2.72541  -0.762  0.44599    
ethnicityAfrican American -0.19974    3.15410  -0.063  0.94952    
ethnicityAsian             2.76124    3.53580   0.781  0.43501    
ethnicityCaucasian        -0.09807    3.05814  -0.032  0.97442    
ethnicityHispanic         -0.74139    3.10275  -0.239  0.81119    
prestigeHigh               1.21437    3.12186   0.389  0.69736    
prestigeLow                3.42243    3.48146   0.983  0.32581    
prestigeMedium            -0.14855    2.02300  -0.073  0.94148    
skill_score                0.11465    0.04528   2.532  0.01148 *  
project_score              0.14796    0.02131   6.942 6.75e-12 ***
experience_score           0.36927    0.02759  13.387  < 2e-16 ***
years_experience          -0.25405    0.08298  -3.061  0.00226 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 9.747 on 1053 degrees of freedom
Multiple R-squared:  0.3121,	Adjusted R-squared:  0.3036 
F-statistic: 36.75 on 13 and 1053 DF,  p-value: < 2.2e-16
```

Table 1: Initial Model Coefficients

Predictor	**Meta β (SE)**	**p-value**	**Microsoft β (SE)**	**p-value**	**IBM β (SE)**	**p-value**
Intercept	93.81 (4.934)	$< 2 \times 10^{-16}$	32.87 (4.896)	3.07×10^{-11}	77.70 (6.487)	$< 2 \times 10^{-16}$
Gender						
Female	-6.32 (4.681)	0.177	-2.409 (2.718)	0.376	-2.25 (2.923)	0.441
Male	-6.08 (4.681)	0.194	-2.078 (2.725)	0.446	-6.902 (2.91)	0.018

Ethnicity						
African American	-0.945 (1.022)	0.355	-0.1997 (3.15)	0.950	8.545 (5.715)	0.135
Asian	-0.343 (1.252)	0.784	2.761 (3.536)	0.435	6.521 (6.675)	0.329
Caucasian	-0.867 (0.960)	0.367	-0.098 (3.058)	0.974	11.31 (5.456)	0.039
Hispanic	0.3980 (0.997)	0.690	-0.741 (3.103)	0.811	10.61 (5.618)	0.059
Prestige						
High	-0.415 (0.650)	0.524	1.214 (3.122)	0.697	-3.686 (2.145)	0.086
Medium	0.1902 (0.541)	0.725	-0.1486 (2.023)	0.941	0.687 (1.786)	0.701
Low	-0.021 (0.516)	0.968	3.422 (3.481)	0.326	-3.462 (1.698)	0.042
Scores						
Skills	0.0007 (0.009)	0.933	0.1147 (0.045)	0.011	-0.1036 (0.05)	0.024
Projects	0.0081 (0.009)	0.345	0.1480 (0.021)	6.75×10^{-12}	-0.0763 (0.04)	0.058
Experience	-0.001 (0.011)	0.932	0.3693 (0.028)	$< 2 \times 10^{-16}$	-0.0804 (0.04)	0.028
Years Experience	0.0168 (0.056)	0.764	-0.2541 (0.083)	0.002	-0.7438 (0.18)	4.73×10^{-5}

Let's interpret all significant predictors at a value of $\alpha = 0.05$ according to the values in Table 1. For the Meta model, there were not any significant predictors, instead only the intercept was significant. Specifically for the Meta model, when all predictors are at 0 and the categorical variables are at their reference group (Unknown), then the average expected score is 93.81. This suggests either the response variable (resume score) is either independent of the predictors or some transformations need to be applied to the response variable and/or the predictors.

For the Microsoft model, the only statistically significant predictors were the numerical ones. In the context of this problem, this is most likely a good thing as it suggests that the Microsoft language model is unbiased towards gender, ethnicity and institutional prestige. For the experience score predictor, the linear regression resulted in a coefficient of 0.3693, suggesting that the resume score increases by that value on average per unit increase of the experience score while holding all other predictors constant. Similarly, the skills score increased the resume score by 0.1147 on average per unit increase of skills score while holding the other predictors constant. Project score also increased the resume score by 0.1480 on average per unit increase of project score while holding the other predictors constant. Most surprising to me was the years experience had a negative coefficient, suggesting that the resume score decreased by 0.2541 per year increase of experience while all other predictors remained constant. This could suggest that the Microsoft model is discriminating against older individuals who are closer to retirement but more data analysis would need to be done.

The IBM model was quite sporadic with its significant predictors. All the categorical variables (gender, prestige, and ethnicity) were significant as well as the skills and experience scores and years experience. For ethnicity, knowing the candidate was white (compared to unknown ethnicity) increased the resume score by an average of 11.31 while holding all other predictors constant. This was also the highest increase for any categorical variable across any model. For gender, knowing the candidate was male (compared to unknown gender) decreased the resume score by an average of 6.902 while holding all other predictors constant. This was the greatest decrease for any categorical predictor across all models. For prestige, knowing the candidate was from a school with less prestige (compared to unknown prestige) decreased the resume score by an average of 3.462 while holding all other variables constant.

Model Improvements and Diagnostics

```
library(onewaytests)

shapiro.test(models$ibm$residuals)
```

```
	Shapiro-Wilk normality test

data:  models$ibm$residuals
W = 0.90642, p-value < 2.2e-16
```

```
shapiro.test(models$meta$residuals)
```

```
	Shapiro-Wilk normality test

data:  models$meta$residuals
W = 0.95486, p-value < 2.2e-16
```

```
shapiro.test(models$microsoft$residuals)
```

```
	Shapiro-Wilk normality test

data:  models$microsoft$residuals
W = 0.9736, p-value = 5.042e-13
```

First I applied the Shapiro-Wilk test to all the models to determine if the normality assumption is satisfied. All models resulted in p-values less than $\alpha = 0.05$ which suggests that none of the residuals are normally distributted. More specifically, the Meta and IBM models had p-values less than 2.2×10^{-16} and the Microsoft model had a p-value of 5.042×10^{-13}. Thus, I will apply a box-cox transformation in an attempt to make the residuals normally distributed.

```
library(faraway)
```

```
library(faraway)

vif(models$ibm)
```

```
          genderFemale             genderMale ethnicityAfrican American
              5.212369               5.176482                  8.290406
        ethnicityAsian     ethnicityCaucasian         ethnicityHispanic
              2.867575              16.238085                 11.216688
          prestigeHigh            prestigeLow            prestigeMedium
              1.363775               1.580460                  1.553167
           skill_score          project_score          experience_score
              1.021806               1.096735                  1.122238
      years_experience
              1.067864
```

```
vif(models$meta)
```

```
          genderFemale             genderMale ethnicityAfrican American
            134.574471             134.580401                  4.044371
        ethnicityAsian     ethnicityCaucasian         ethnicityHispanic
              2.001009               5.629096                  4.447988
          prestigeHigh            prestigeLow            prestigeMedium
              1.214805               1.301816                  1.273200
           skill_score          project_score          experience_score
              1.007023               1.019463                  1.016656
      years_experience
              1.014186
```

```
vif(models$microsoft)
```

```
          genderFemale             genderMale ethnicityAfrican American
             20.732645              20.855516                 12.349097
        ethnicityAsian     ethnicityCaucasian         ethnicityHispanic
              3.837145              24.115571                 16.783707
          prestigeHigh            prestigeLow            prestigeMedium
              1.016330               1.013075                  1.010703
           skill_score          project_score          experience_score
              1.029190               1.388214                  1.400930
      years_experience
              1.034506
```

Next according to the VIF test, the IBM model had VIF's greater than 10 for both the Caucasian (16.24) and Hispanic (11.22), suggesting that multicolinearity is present in the IBM model. For the Meta model, the Gender variable had VIFs greater than 10 (both Male and Female had VIFs greater than 134) suggesting the Meta model may also be subject to multicolinearity. Lastly the microsoft model had VIFs of over 20 for the gender column and many ethnicity categories a lso had VIFs greater than 10. Thus all models were subject to multicolinearity and will use Ridge Regression to help mitigate the effects of multicolinearity.

```r
gender_ethnicity_models = NA
gender_ethnicity_models$ibm = lm(score ~ gender*ethnicity + gender + ethnicity + prestige + ski
gender_ethnicity_models$meta = lm(score ~ gender*ethnicity + gender + ethnicity + prestige + sk
gender_ethnicity_models$microsoft = lm(score ~ gender*ethnicity + gender + ethnicity + prestige

anova(models$ibm, gender_ethnicity_models$ibm, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ gender * ethnicity + gender + ethnicity + prestige +
    skill_score + project_score + experience_score + years_experience
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    838 291396
2    831 287987  7    3408.9 1.4052 0.1997
```

```r
anova(models$meta, gender_ethnicity_models$meta, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ gender * ethnicity + gender + ethnicity + prestige +
    skill_score + project_score + experience_score + years_experience
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1   1042 44771
2   1037 44753  5     17.63 0.0817 0.9951
```

```r
anova(models$microsoft, gender_ethnicity_models$microsoft, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ gender * ethnicity + gender + ethnicity + prestige +
    skill_score + project_score + experience_score + years_experience
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   1053 100030
2   1047  99263  6    767.23 1.3488 0.2325
```

```r
gender_presitge_models = NA
gender_presitge_models$ibm = lm(score ~ gender*prestige + gender + ethnicity + prestige + skill
gender_presitge_models$meta = lm(score ~ gender*prestige + gender + ethnicity + prestige + skil
gender_presitge_models$microsoft = lm(score ~ gender*prestige + gender + ethnicity + prestige +

anova(models$ibm, gender_presitge_models$ibm, test="F")
```

```
Analysis of Variance Table
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ gender * prestige + gender + ethnicity + prestige + skill_score +
    project_score + experience_score + years_experience
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    838 291396
2    832 290837  6    558.83 0.2664 0.9525
```

```
anova(models$meta, gender_presitge_models$meta, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ gender * prestige + gender + ethnicity + prestige + skill_score +
    project_score + experience_score + years_experience
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1   1042 44771
2   1038 44678  4    92.478 0.5371 0.7085
```

```
anova(models$microsoft, gender_presitge_models$microsoft, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ gender * prestige + gender + ethnicity + prestige + skill_score +
    project_score + experience_score + years_experience
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   1053 100030
2   1050  99927  3    102.76 0.3599  0.782
```

```
ethnicity_presitge_models = NA
ethnicity_presitge_models$ibm = lm(score ~ prestige*ethnicity + gender + ethnicity + prestige +
ethnicity_presitge_models$meta = lm(score ~ prestige*ethnicity + gender + ethnicity + prestige
ethnicity_presitge_models$microsoft = lm(score ~ prestige*ethnicity + gender + ethnicity + pres

anova(models$ibm, ethnicity_presitge_models$ibm, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ prestige * ethnicity + gender + ethnicity + prestige +
    skill_score + project_score + experience_score + years_experience
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    838 291396
```

```
2    826 286938 12    4457.7 1.0693  0.383
```

```
anova(models$meta, ethnicity_presitge_models$meta, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ prestige * ethnicity + gender + ethnicity + prestige +
    skill_score + project_score + experience_score + years_experience
  Res.Df   RSS Df Sum of Sq      F Pr(>F)
1   1042 44771
2   1031 44631 11    139.19 0.2923 0.9874
```

```
anova(models$microsoft, ethnicity_presitge_models$microsoft, test="F")
```

```
Analysis of Variance Table

Model 1: score ~ gender + ethnicity + prestige + skill_score + project_score +
    experience_score + years_experience
Model 2: score ~ prestige * ethnicity + gender + ethnicity + prestige +
    skill_score + project_score + experience_score + years_experience
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1   1053 100030
2   1045  99621  8    408.61 0.5358 0.8301
```

To test the presence and significance of interaction effects, I fitted new models on each of the datasetes with the same formula mentioned in section 2.3 with the addition of the interaction term. Specifically the interaction terms added were Ethnicity$\times$Gender, Gender$\times$Prestige, and Prestige$\times$Gender. Each of these interaction terms were added one at a time and compared to the original formula using an anova test to determine if the interaction term was significant. All anova tests had p-values greater than $\alpha = 0.05$ for each interaction term in all of the models so none of the interaction effects were significant and will not be added to the final modeling process.

Improved Model

```
library(MASS)
range(ibm$score)
```

```
[1]   0 100
```

```
ibm$score_new = ibm$score + 0.0000001
range(meta$score)
```

```
[1] 65 98
```

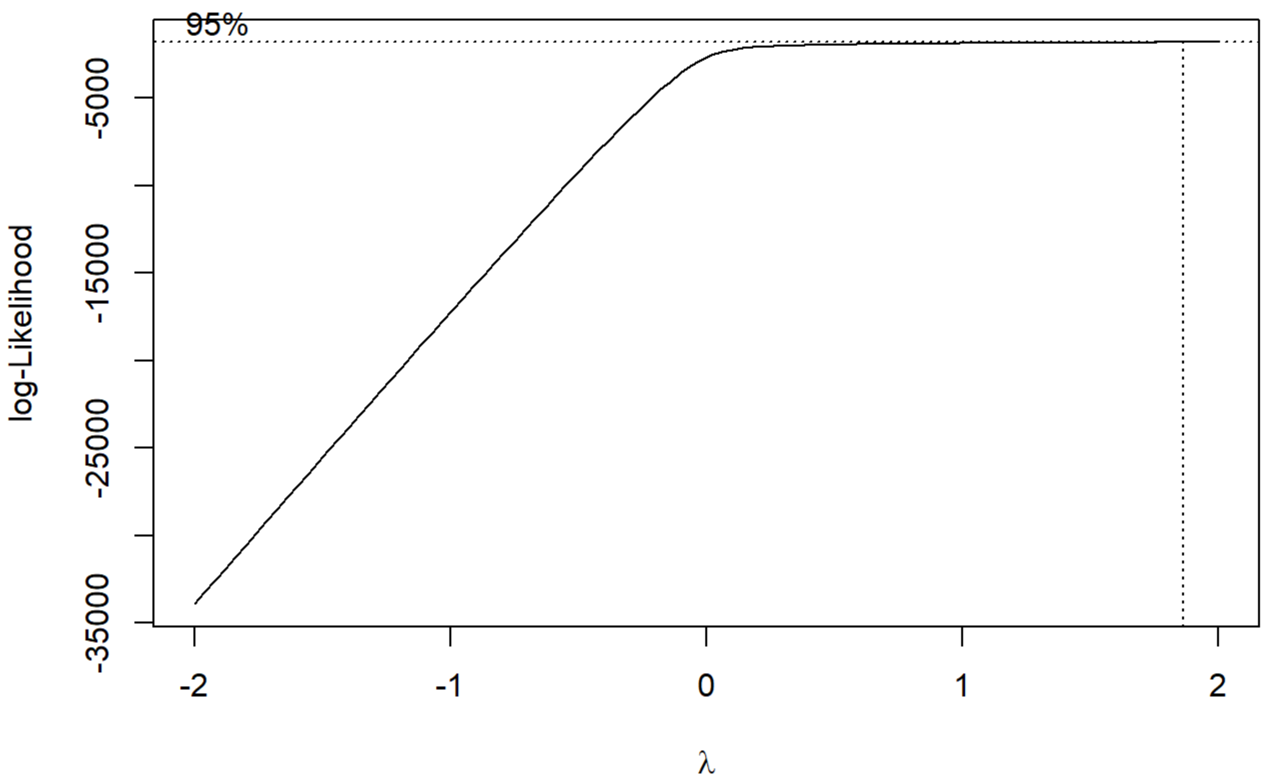
```
range(microsoft$score)
```

```r
range(microsoft$score)
```

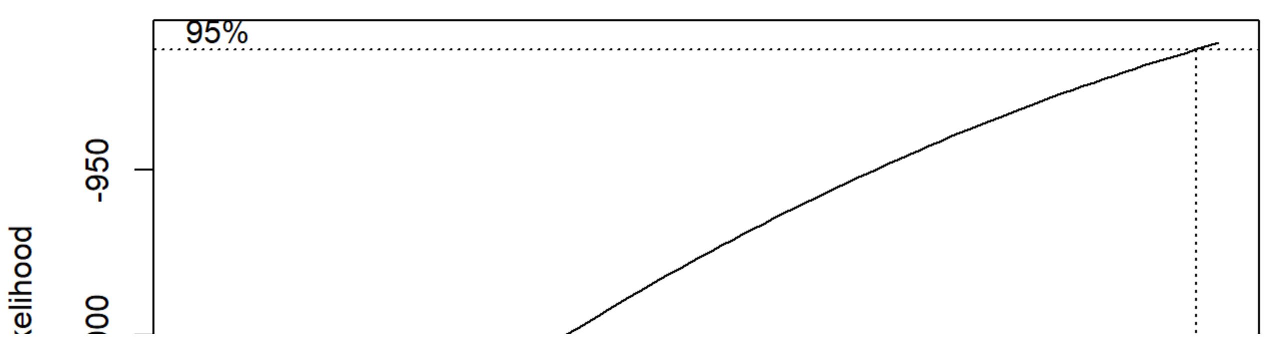
```
[1] 15 85
```

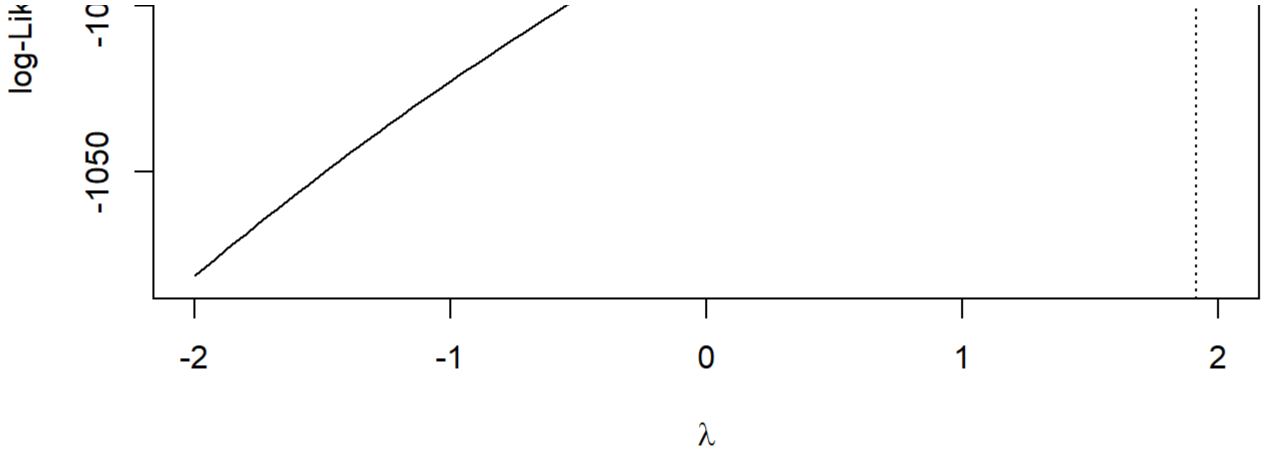
```r
bc = NA

bc$ibm = boxcox(lm(ibm$score_new ~ 1))
```

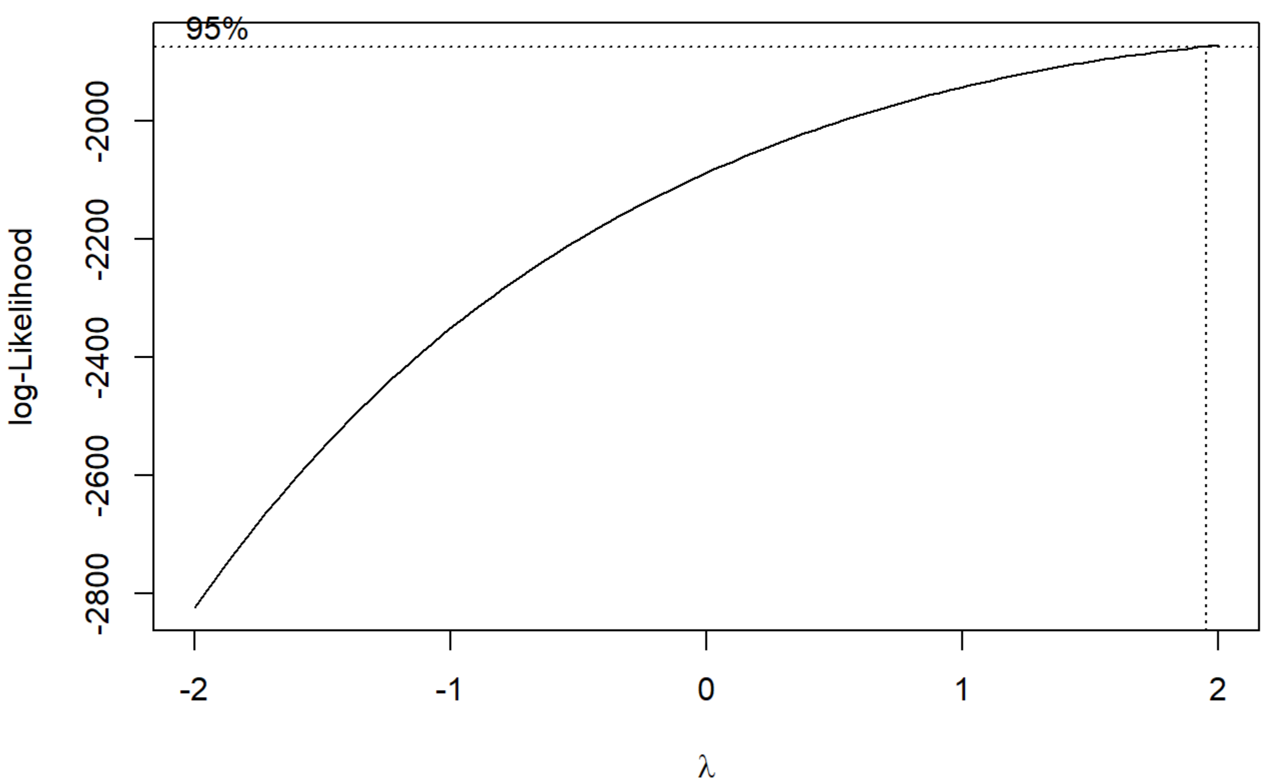



```r
bc$meta = boxcox(lm(meta$score ~ 1))
```






```
bc$microsoft = boxcox(lm(microsoft$score~ 1))
```




```
lambda = NA
lambda$ibm <- bc$ibm$x[which.max(bc$ibm$y)]
lambda$meta <- bc$meta$x[which.max(bc$meta$y)]
lambda$microsoft <- bc$microsoft$x[which.max(bc$microsoft$y)]

print(lambda$ibm)
```

```
print(lambda$ibm)
```

```
[1] 2
```

```
print(lambda$meta)
```

```
[1] 2
```

```
print(lambda$microsoft)
```

```
[1] 2
```

```
X = NA

X$ibm = subset(ibm, select=-c(score, name, gender, ethnicity, prestige, score_new))
X$meta = subset(meta, select=-c(score, name, gender, ethnicity, prestige))
X$microsoft = subset(microsoft, select=-c(score, name, gender, ethnicity, prestige))

Cat = NA

Cat$ibm = subset(ibm, select=c(gender,ethnicity,prestige))
Cat$meta = subset(meta, select=c(gender,ethnicity,prestige))
Cat$microsoft = subset(microsoft, select=c(gender,ethnicity,prestige))

Y = NA

Y$ibm = I(ibm$score^2)
Y$meta = I(meta$score^2)
Y$microsoft = I(microsoft$score^2)

Y$ibm = scale(Y$ibm, center = T, scale = T)
Y$meta = scale(Y$meta, center = T, scale = T)
Y$microsoft = scale(Y$microsoft, center = T, scale = T)

X$ibm <- scale(X$ibm, center = T, scale=T)  ## always remember to standardize X
X$meta <- scale(X$meta, center = T, scale=T)  ## always remember to standardize X
X$microsoft <- scale(X$microsoft, center = T, scale=T)  ## always remember to standardize X

X$ibm = cbind(X$ibm, Cat$ibm)
X$meta = cbind(X$meta, Cat$meta)
X$microsoft = cbind(X$microsoft, Cat$microsoft)

X$ibm <- model.matrix(~ . - 1, data = X$ibm)
X$meta <- model.matrix(~ . - 1, data = X$meta)
X$microsoft <- model.matrix(~ . - 1, data = X$microsoft)

library(glmnet)

opt_lambda <- NA
```

```
par(mfrow = c(1, 3), mar = c(3, 4, 2, 1), oma = c(0, 0, 2, 0))

plot_cv_custom <- function(cv_fit, title, color) {
  lambda <- cv_fit$lambda
  log_lambda <- log(lambda)
  cvm <- cv_fit$cvm  # Mean MSE
  cvsd <- cv_fit$cvsd  # Standard deviation

  upper_bound <- cvm + 2 * cvsd
  lower_bound <- cvm - 2 * cvsd

  ylims <- range(c(lower_bound, upper_bound))

  plot(1, type = "n",
       xlim = range(log_lambda),
       ylim = ylims,
       xlab = "log(λ)",
       ylab = "Mean-Squared Error",
       main = title,
       cex.main = 1.2,
       font.main = 2,
       col.main = color)

  # Add ±2 SD bands as dashed lines
  lines(log_lambda, upper_bound, lty = 2, col = "gray60", lwd = 1.5)
  lines(log_lambda, lower_bound, lty = 2, col = "gray60", lwd = 1.5)

  # Add mean MSE line (solid, thicker)
  lines(log_lambda, cvm, lwd = 3, col = color)

  # Add optimal lambda vertical line
  abline(v = log(cv_fit$lambda.min), lty = 3, col = "red", lwd = 2)
}

cv_fit_ibm <- cv.glmnet(X$ibm, Y$ibm, alpha = 0)
opt_lambda$ibm <- cv_fit_ibm$lambda.min
plot_cv_custom(cv_fit_ibm, "IBM Cross-Validation Results", "darkblue")

cv_fit_meta <- cv.glmnet(X$meta, Y$meta, alpha = 0)
opt_lambda$meta <- cv_fit_meta$lambda.min
plot_cv_custom(cv_fit_meta, "Meta Cross-Validation Results", "darkgreen")

cv_fit_microsoft <- cv.glmnet(X$microsoft, Y$microsoft, alpha = 0)
opt_lambda$microsoft <- cv_fit_microsoft$lambda.min
plot_cv_custom(cv_fit_microsoft, "Microsoft Cross-Validation Results", "darkred")

mtext("Ridge Regression Cross-Validation Results",
      outer = TRUE, cex = 1.5, font = 2, line = 0.5)
```

```
outer = TRUE, cex = 1.5, font = 2, line = 0.5)
```

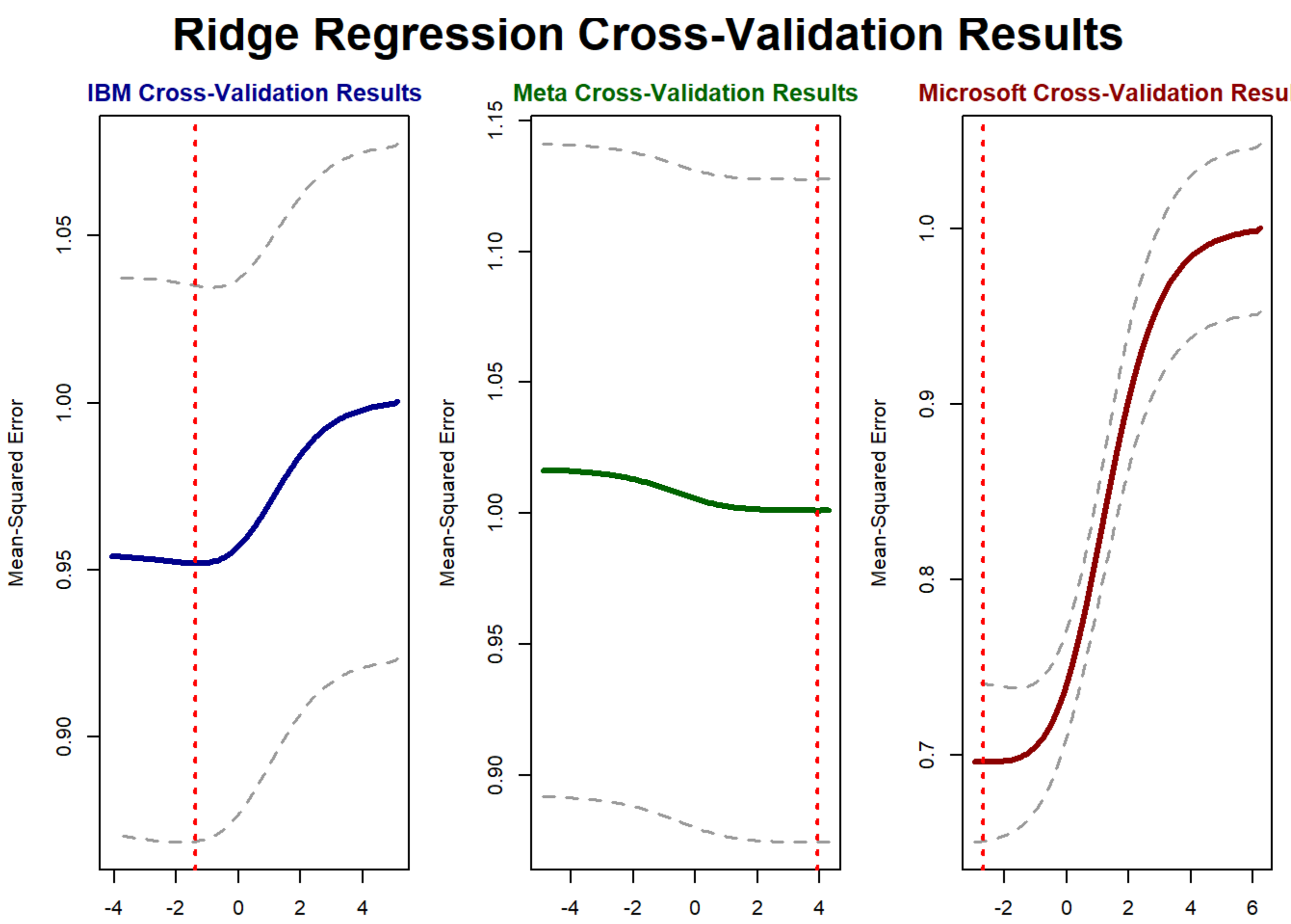


Figure 1: A line plot

When performing the box-cox transformation, all models resulted in $\lambda = 2$ which will then be used to transform the response for the ridge regression. Thus, all the ridge regressions will be performed on Score^2 as the response variable. Note this $\lambda = 2$ refers to the exponent for our response variable and not the weight of the penalty term in the ridge regression. It is also important to center and scale all numerical predictors during ridge regression because ridge regression is sensitive to the scale of inputs.

Next was performing Cross-Validation for the ridge regression to find the optimal parameter for the penalty term. According to Figure 1, the IBM model had an optimal lambda of 0.2498. For Meta, the optimal lambda was 55.98. For Microsoft, the lambda was 0.0514.

```r
ridge_models = NA

print(opt_lambda)
```

```
[[1]]
[1] NA

$ibm
[1] 0.2498869
```

```
$meta
[1] 51.00987

$microsoft
[1] 0.06797179
```

```
ridge_models$ibm = glmnet(X$ibm, Y$ibm, alpha=0, lambda= opt_lambda$ibm)
coef(ridge_models$ibm)
```

```
15 x 1 sparse Matrix of class "dgCMatrix"
                                    s0
(Intercept)                -0.003652022
skill_score                -0.059098278
project_score              -0.055251286
experience_score           -0.058690211
years_experience           -0.123047145
genderUnknown               0.166254465
genderFemale                0.076351439
genderMale                 -0.109766313
ethnicityAfrican American -0.032425989
ethnicityAsian             -0.112957753
ethnicityCaucasian          0.088203530
ethnicityHispanic           0.054870354
prestigeHigh               -0.121943195
prestigeLow                -0.137562050
prestigeMedium              0.049074274
```

```
ridge_models$meta = glmnet(X$meta, Y$meta, alpha=0, lambda= opt_lambda$meta)
coef(ridge_models$meta)
```

```
15 x 1 sparse Matrix of class "dgCMatrix"
                                     s0
(Intercept)                4.221920e-04
skill_score                6.926210e-05
project_score              4.841967e-04
experience_score           6.145402e-05
years_experience           3.478145e-04
genderUnknown              1.747399e-02
genderFemale              -1.073290e-04
genderMale                -2.484210e-05
ethnicityAfrican American -1.438408e-03
ethnicityAsian             5.323934e-04
ethnicityCaucasian        -1.913997e-03
ethnicityHispanic          3.303882e-03
prestigeHigh              -1.295985e-03
prestigeLow               -3.137222e-05
prestigeMedium             5.836672e-04
```

```
ridge_models$microsoft = glmnet(X$microsoft, Y$microsoft, alpha=0, lambda= opt_lambda$microsoft
```

```
ridge_models$microsoft = glmnet(X$microsoft, Y$microsoft, alpha=0, lambda= opt_lambda$microsoft
coef(ridge_models$microsoft)
```

```
15 x 1 sparse Matrix of class "dgCMatrix"
                                   s0
(Intercept)                -0.0045527081
skill_score                 0.0746989212
project_score               0.2197804319
experience_score            0.3758506147
years_experience           -0.0712203340
genderUnknown               0.1518195412
genderFemale               -0.0060123115
genderMale                 -0.0005520487
ethnicityAfrican American -0.0097338411
ethnicityAsian              0.2564995872
ethnicityCaucasian          0.0103574875
ethnicityHispanic          -0.0491372663
prestigeHigh                0.0757202988
prestigeLow                 0.2687122307
prestigeMedium             -0.0058210648
```

After finding the optimal lambda for each model, we can report the 4 largest coefficients, which represent the variables most important for explaining the variation of the response variable. For the IBM model, the 4 largest coefficients were low institutional prestige (-0.1376), high institutional prestige (-0.1219), years experience (-0.1230) and asian ethnicity (-0.1129). For the meta model, all coefficients were extremely small suggesting a rather poor fit overall. Specifically, the largest were hispanic ethnicity (3.018×10^{-3}), caucasian ethnicity (\(-1.748\times 10^{-3\})), african american ethnicity (-1.312×10^{-3}) and high institutional prestige (-1.183×10^{-3}). While these models are poor fits ($R^2_{ibm} = 0.0689, R^2_{meta} = 0.00042$) this still demonstrates subtle biases within these models which should aim to be removed. This demonstrates this because the most important factors for the ridge regression were the categorical variables (gender, prestige, ethnicity) that should not influence a candidates' fit for the job.

The microsoft model had by far the best results for minimizing bias and also the highest performing model of the bunch ($R^2_{microsoft} = 0.3221$). The top 4 largest predictors for this model were experience score (0.3817), low prestige (0.2705), asian ethnicity ($ 0.2625$), and project score ($0.2203$), with most other predictors being below 0.1. It is important to note when interpreting the coefficient for a ridge regression, we are looking at an average increase in the response (which is a scaled and centered Score^2) per unit increase in the predictor (for numerical varaibles a unit would be one standard deviation since the variables are scaled, for categorical interpretation is the same as before) while holding the other variables constant.

```
#use fitted best model to make predictions
y_predicted <- predict(ridge_models$ibm, s = opt_lambda$ibm, newx = X$ibm)

#find SST and SSE
sst <- sum((Y$ibm - mean(Y$ibm))^2)
sse <- sum((y_predicted - Y$ibm)^2)

#find R-Squared
```

```
rsq <- 1 - sse/sst

print("ibm")
```

```
[1] "ibm"
```

```
rsq
```

```
[1] 0.06898701
```

```
summary(models$ibm)$r.squared
```

```
[1] 0.0767305
```

```
#use fitted best model to make predictions
y_predicted <- predict(ridge_models$meta, s = opt_lambda$meta, newx = X$meta)

#find SST and SSE
sst <- sum((Y$meta - mean(Y$meta))^2)
sse <- sum((y_predicted - Y$meta)^2)

#find R-Squared
rsq <- 1 - sse/sst

print("meta")
```

```
[1] "meta"
```

```
rsq
```

```
[1] 0.0004594969
```

```
summary(models$meta)$r.squared
```

```
[1] 0.01016942
```

```
#use fitted best model to make predictions
y_predicted <- predict(ridge_models$microsoft, s = opt_lambda$microsoft, newx = X$microsoft)

#find SST and SSE
sst <- sum((Y$microsoft - mean(Y$microsoft))^2)
sse <- sum((y_predicted - Y$microsoft)^2)

#find R-Squared
rsq <- 1 - sse/sst
print("microsoft")
```

```
[1] "microsoft"
```

```
rsq
```

```
[1] 0.3218248
```

```
summary(models$microsoft)$r.squared
```

```
[1] 0.3121289
```

Conclusion

Thus, this project found that there is likely bias in the decision making process of resume screeners. More specifically, for most of the models, gender, ethnicity, and institutional prestige seemed to play more of an important role in determining a candidate's fit for the job than other professional factors like work experience and projects. Additionally, the data was extremely variable with each model having completely different distributions of scores, ethnicitiies and prestiges, with certain models not appearing to have a very consistent scoring process or method. It is important to note however that the model that utilized the professional experience the most in determining the resume's overall score was the Microsoft model (*phi4_14b*) which also had the largest parameter size of the group. More studies should be done to analyze if model size plays a factor in presence of bias as well as potentially look at a temporal aspect by seeing if newer models tend to have less biases than older models.

Footnotes

1. https://www.demandsage.com/ai-recruitment-statistics/ ↩
2. https://www.washington.edu/news/2024/10/31/ai-bias-resume-screening-race-gender/ ↩
3. https://huggingface.co/datasets/datasetmaster/resumes ↩