Proving gender bias in the labor market

Data Science in Action Winter Project

Claudia Imperatore, Matteo Gioia and Vincenzo Junior Striano

INTRODUCTION

The impact of technology on our lives is growing rapidly. Artificial Intelligence algorithms are applied daily in different fields: in the medical field, in self-driving vehicles, to determine if we are worthy of a mortgage or to determine if we deserve a certain job position. An error in an algorithm might not seem very pleasant to us, but not even that serious: if Netflix recommends a film we don't like or if Siri sets the alarm clock at the wrong time, we are faced with trifles that we could turn a blind eye to, given the countless facilities they offer us. But what if the error concerned an autonomous driving algorithm? Or if we were rejected at a job interview because of our gender, religion, or race?

The mistake that has probably caused the most stir in recent years is related to the recruitment software algorithm used by Amazon starting in 2014. This software was believed to analyze candidates' resumes and automate the selection process. However, it emerged how it penalized women, especially for positions related to more technological roles. The error was due to the data with which the model was trained: real data, containing the resumes received by the company in the previous 10 years; purely male resumes, given the majority of men in the technology sector. The model automatically recognized a pattern that delineated the best candidates, incorporating the male gender among the ideal characteristics, thus incurring a bias. A bias is a systematic error in judgment or interpretation, which can lead to a misjudgment or to making an unobjective judgment. It is a form of cognitive bias caused by prejudice and can influence ideologies, opinions, and behaviors. In computer science, algorithmic bias is an error due to incorrect assumptions in the machine learning process. This error forced Amazon to decommission the software.

The previously reported errors occur because by training AI models through the massive amounts of data available to us, AI incorporates values and biases inherent in society. Although the common imagination leads us to consider an algorithm as a perfect decision-making process, superior to human reasoning (considered instead as biased and not objective), because it can process a multiplicity of data in an unbiased way, in reality, this is not the case.

Algorithmic fairness is a growing field of research that aims to mitigate the effects of bias and unjustified discrimination on individuals in machine learning, primarily focused on mathematical formalism and finding solutions for these formalisms. It is an interdisciplinary research area that aims to create learning models capable of making fair predictions from the perspective of equity and justice.

As future data scientists, we're interested to know whether in our class someone is affected by gender bias, which can be poured into our future projects.

In particular, we want to investigate more about what the Harvard Business Review website spoke about in 2014*: Men apply for a job when they meet only 60% of the qualifications, but women apply only if they meet 100% of them*.

Our goal is to test whether this assumption is actually true. If it turns out to be true, any algorithm influenced by this bias would produce a biased system, which with a large database would simply automate the error and standardize it by favoring the assumptions of one sex over the other.

PREMISE TO THE EXPERIMENT

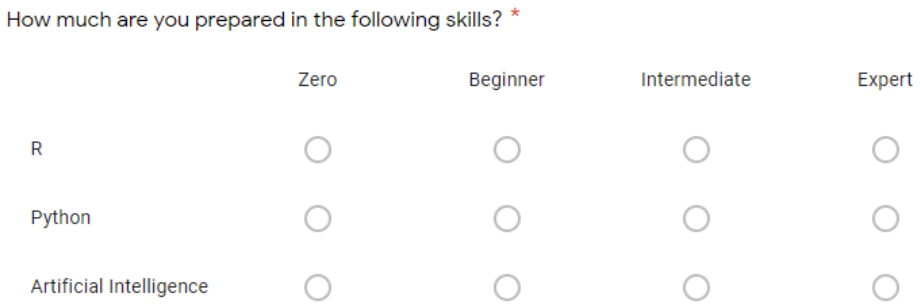
The aim of this winter project was to carry out a mock version of an actual Data Science project. Despite the teams 'efforts, the project could not be carried out successfully. This was because a short number of observations have been collected, resulting in the impossibility of testing the models. Despite numerous requests, a small group of 8 individuals completed our survey. This was not enough to reach statistically significant conclusions. Regardless, determined to carry out our work as best as possible, we transformed a simple descriptive paper into a didactic guide on how to carry out the experiment we had in mind in an ideal scenario, that is to say, having the chance to work with a higher number of observations. Therefore, it was necessary to change part of the structure in the work: conclusions cannot be drawn due to the lack of data. However, we, therefore, focused on how to create a simple and error-free model and show what all possible outputs are. At the end of the work, an appendix is available which shows the results, albeit statistically insignificant, of the model based on the data available to us.

METHODS

Data collection

For the data collection step, we created a Google Form survey divided into 2 main sections. We surveyed our colleagues of the Data Science and Management class, predominantly Italian students (screenshot of the survey available at https://is.gd/surveyCMV).

The first section aimed to identify the gender of each individual and to collect the level of preparation regarding a list of hard and soft skills that will be shown in the second step. This step allows us to get feedback on how prepared individuals feel from "Zero" to "Expert" on the various skills so we can plot each value between 1 and 4. Each skill level will allow us to compare each individual's preparation with the preparation required in the various job postings, from phase 2 of our survey, and get feedback on the different propensity to apply to job postings between men and women. We'll back on these in the *Assumption* chapter.



In the second section, we submitted 5 job advertisements to individuals for the role of "Junior Data Scientist" in a 6-month internship.

Each job advertisement consists of the title, a brief description of the vacant position, the general characteristics that the role requires and the requirements A and B that correspond to Soft and Hard Skills respectively. The advs differs **only** in the requirements where different skills and levels required by the position are indicated that correspond to the same ones analyzed in section 1 of our survey, to lower the chance of any other conditioning (see the picture on the next page).

After the individual has analyzed an advertisement, they are asked to answer whether or not they would apply for the position in question.

Would you apply to this job? *

- ◯ Yes
- ◯ No

JUNIOR DATA SCIENTIST INTERNSHIP, COMPANY "ONE"
Internship, 6 months

About the job

Are you a student in a STEM subject, Data Science or something and looking to get some work experience under your belt? Itching to get your hands into a tech team where processes are still malleable, and you can really make a difference?

You are?

Great!

Your role

- Working with statistical programming language to create unsupervised predicting model for our business
- Daily present your colleagues, supervisors, and CEO your progress

Requirements A:

- Intermediate ability to speak publicly
- Expert ability to work on their own

Requirements B:

- Expert knowledge of R
- Intermediate use of Excel
- Basic knowledge of Tableau software for data visualization
- Intermediate knowledge of unsupervised learning

FEATURE ENGINEERING

Thanks to the survey, we can retrieve the values of our predictors and the target variable.

The target variable. through which it will be possible to understand whether it is present a bias or not, is answered to “Would you apply to this job positions?”. If the gender bias is confirmed, women will apply to the job advertisements only when they meet 100% of the requirements.

How much are you prepared in the following skills? [Artificial Intelligence]	How much are you prepared in the following skills? [Public Speaking]	How much are you prepared in the following skills? [Autonomous
0	3	2
2	2	3
0	2	0
1	2	2
0	2	2
0	2	3
1	2	2
1	1	2
1	3	3

To build the model, it’s important to introduce further variables that will help in understanding whether the hypothesis is confirmed or not.

Need variables

Each requirement (eg. Python) is associated with a verbal level which can be Basic, Intermediate, or Expert. Each verbal level corresponds to 1, 2, or 3. The need variables summarize the average level of competencies required for each class of requirements in the advertisements. Let’s make an example:

The first job advertisements will have two need variables: NEEDA and NEEDB, one for each class of requirements.

Let’s compute the value of NEEDA: Intermediate (2) ability to (…) + Expert (3) ability to work…

NEEDA = (2+3)/2 = 2.

Requirements A:

- Intermediate ability to speak publicly
- Expert ability to work on their own

Requirements B:

- Expert knowledge of R
- Intermediate use of Excel
- Basic knowledge of Tableau software for data visualization
- Intermediate knowledge of unsupervised learning

Avg variables

The avg variables compute the average level of competencies of the individual required for the job adv. Each competence is declared by the individual at the beginning of the survey and the values can range between 0 and 4 (from Zero to Expert). They are computed through a simple average for each section A and B accordingly to the skills required. Let's look again at the figure above: in this case, the skills of interests are going to be public speaking and autonomous working for the class Requirements A. Therefore, AVGA is computed by the average between the level of competencies about public speaking and autonomous working.

Diff variables

The *diff* variables explain how much each individuals' level matches with the level of skills required by the job adv, so it's computed by the ratio of AVG/NEED. If the hypothesis holds, in an ideal scenario, there will be a threshold of *diff* below which people won't apply. Above that threshold, males will tend to apply more than women. When *diff* tends to 100% the difference between male and women will lower. Only when *diff* = 100% the difference between males and females will reach its minimum.

DIFF	SEX	
	M	F
0-10	0	0
…		
60-70	100,0%	0,0%
70-80	87,5%	12,5%
80-90	75,0%	25,0%
90-100	62,5%	37,5%
100	50,0%	50,0%

Example of "diff" distribution in presence of a bias

ASSUMPTIONS

Before introducing the model, it is important to state that the main assumption of our experimental design is that there is no difference in weight between hard and soft skills and between each of the skills, either from an applicant or recruiter's point of view. This is the reason why simple averages were used in the computation of the variables. Furthermore, we assume that the participants of the survey can discern the differences in the level of their preparation. We acknowledge that in the future, the survey might be improved by directly testing users.

MODEL CHOICE

As stated before, the dependent variable of the model is going to be the variable 'Would you apply to this job?'. The answer might be 'yes' or 'no'. Considering that the output of the variable is binary and that the problem is a problem of classification, we have identified two different models that might be used, one of which can be used as a benchmark: Logistic Regression and Support Vector Classification. We are going to apply the model to each of the job adv. In the end, the models are summarized performing an average of the built hyperparameters for each job adv.

For both models, the independent variables are DIFF -which summarize need, avg and users' skill- and GENDER. The dependent variable is "Would you apply to this job adv? Y/N". Python 3 will be used as a programming language.

Logistic regression

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. It uses a log of odds as the dependent variable.

Support Vector Machine – Classifier

Support Vector Machine (SVM) constructs a bidimensional hyperplane to separate different classes and minimize errors. The main objective is to segregate the given dataset in the best possible way. The distance between the either nearest point is known as the margin. The objective is to select a hyperplane with the maximum possible margin between support vectors in the given dataset. SVM searches for the maximum marginal hyperplane in the following steps:

1. Generate hyperplanes that segregate the classes in the best way.
2. Select the right hyperplane with the maximum segregation from the either nearest data points.

A confusion matrix will be used to assess the predictive performance and count true and false positives and negatives. Furthermore, measures of accuracy, precision, and recall will be used

THE GENDER BIAS

To understand if the gender variable influences the results, and therefore to understand whether a gender bias is shown in data, we would suggest applying the models on the data twice: one time introducing the gender variable and one time not. At the end of the process, a comparison between accuracy, precision, and recall is performed. If gender influences the dependent variable, the models with gender are going to show a better measure of accuracy, precision, and recall, considering that more information is fed into the models.

APPENDIX – DEMONSTRATION AND CODE EXPLANATION

We apply the model described above to the data we have collected in the manner indicated in the appropriate section.

1. The database generated by the survey responses can be downloaded directly from Google Form. We focus the analysis on the first 21 variables that represent the score from 0 to 3 on how well each individual is prepared on the requirements of the advertisements that will be shown later..
2. We import the *pandas* library of which we execute the *read_csv* command to import the file into the Python environment and transform it into a dataframe. We specify the arguments *sep = ';'* -that indicates the type of data separator- and *header = None*.
   The last argument allows us to get around the problem of particularly complex column headings that would make the code difficult to read every time it is necessary to recall the column itself: the column header becomes a real observation that thanks to the drop command are eliminated.
   Thanks to the *astype (float)* command, "numeric" strings are transformed into decimal numbers. With *reset_index ()*, row 0, deleted with the drop operation, returns to index the first observation relative to the scores. This operation is necessary to allow the correct merging of multiple data frames.

```
import pandas as pd
dfa1 = pd.read_csv('filepreprocessato.csv', sep = ';', header = False)
dfa1 = dfa1.drop(0)
dfa1 = dfa1.astype(float)
dfa1 = dfa1.reset_index()
```

1	2	3	4
1.0	1.0	0.0	3.0
3.0	3.0	2.0	2.0
1.0	1.0	0.0	2.0

3. The data is manipulated, introducing the variables necessary for the analysis. To make the script lighter, the *numpy* library is used whenever possible. In particular, the *repeat* function, which repeats the first argument as many times as you specify in the second, and array which generates a list (or rather, a numpy array). The other variables are obtained from linear combinations of others. All variables are placed in a dataframe. Note the *.transpose ()* argument which generates the transposed dataframe *dfa2*. This is because the *DataFrame* command generates a dataframe by stacking the arrays and not chaining them horizontally.

```
import numpy as np
needa1 = np.repeat(2.55,9)
needb1 = np.repeat(2,9)
avga1 = dfa1[[4, 5]].mean(axis=1)
avgb1 = dfa1[[1, 7, 9, 12]].mean(axis=1)
diffa1 = avga1/needa1
diffb1 = avgb1/needb1
q1 = np.array([0.0,1.0,0.0,0.0,0.0,1.0,1.0,0.0,1.0])
dfa2 = pd.DataFrame((needa1, needb1, avga1,
                avgb1, diffa1, diffb1, q1)).transpose()
```

4. Step 3 is repeated per each advertising.

5. The dataframes are concatenated through the *concat* command, specifying the *axis = 1* which concatenates the dataframes horizontally, not maintaining the column indexes of the original dataframes thanks *to ignore_index = False*. Numeric values are converted back to decimal.

```
dfa1 = pd.concat([dfa1,dfa2,dfa3,
                  dfa4,dfa5,dfa6],
                  axis = 1, ignore_index=True)
```

21	22	needa1	needb1
1.0	2.0	2.55	2.0
3.0	2.0	2.55	2.0
3.0	2.0	2.55	2.0

6. For the demonstration, **a Support Vector Classificator** model is created, whose predictors will be *diffa1*, *diffa2,* and the target variable *Y*, which represents the choice, or not, to send your application to the first advertisement. Thanks to the *iloc* command we create the variable X and the variable Y, selecting the variables by their column indexes.

```
X = dfa1.iloc[:, [27,28]].values
Y = dfa1.iloc[:, 29].values
```

7. We import the *train_test_split* command from the *sk.learn.model_selection* library thanks to which we divide the randomly ordered database into its first 75th part, dedicated to model training, and the last 25th part, dedicated to testing. The *random_state* command allows you to replicate the randomization.

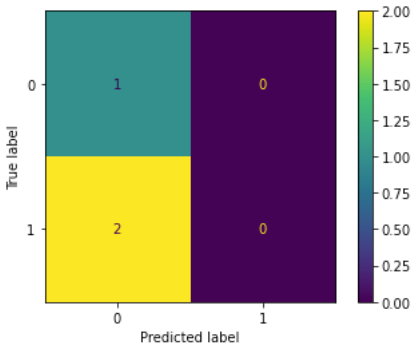
```
from sk.learn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                                                    train_size = 0.75,
                                                    random_state=42)
```

8. We import the *SVC* function from the *sklearn.svm* library. The classifier will have a *linear kernel*, so we fit the model to the available data. We make the predictions of the y, with the hyperparameters just made, based on the X_tests.

```
from sklearn.svm import SVC
classifier = SVC(kernel = 'linear', random_state = 0)
classifier.fit(X_train, y_train)
y_pred = classifier.predict(X_test)
```

9. The *confusion matrix* is displayed thanks to the special function in the *sk.learn.metrics* library. Each column of the matrix represents the predicted values, each row the real values. The element on row i and column j is the number of cases in which the classifier has classified the "true" class i as class j. Through this matrix, it is observable if there is "confusion" in the classification of different classes.

```
from sklearn.metrics import confusion_matrix
cm = confusion_matrix(y_test, y_pred)
from sklearn.metrics import ConfusionMatrixDisplay
cm_display = ConfusionMatrixDisplay(cm).plot()
```



10. We measure the accuracy, precision, and recall of predictions through the specific functions of the *sklearn.metrics* library.

```
from sklearn.metrics import accuracy_score, precision_score,
recall_score
print(accuracy_score(y_test,y_pred))
print(precision_score(y_test,y_pred))
print(recall_score(y_test,y_pred))
```

```
0.0
0.0
0.0
```

11. For the sake of completeness, this time we build a **Logistic Regression** model, inserting the gender of the respondents among the predictors. The procedure linked to the division between train and test, confusion matrix, measurement of predictive performance is identical. The only differences are to be found in the choice of variables and the model.

```
X = dfa1.iloc[:, [27,28,58]].values
y = dfa1.iloc[:, 29].values

from sklearn.linear_model import LogisticRegression
clf = LogisticRegression(random_state=0).fit(X_train, y_train)
```

CONCLUSIONS

As previously mentioned, making conclusions based on the data we have is impossible. However, the team has built the experiment and the model following all the guidelines provided in the lectures.
If the hypothesis is confirmed, and therefore it is shown that, given the same level of skills, men will apply more frequently than women, adding the variable gender to the "skill model" will significantly increase its predictive performance.
To remove the bias, two types of solutions can be provided: besides crucial inclusiveness policies, from a statistical approach, it is possible to deal with a bigger amount of data, balance the data among females and males, let the model be trained by people with diverse backgrounds, and measuring predictive performances for each demographic category.

BIBLIOGRAPHY

Harvard Business Review,
https://hbr.org/2014/08/why-women-dont-apply-for-jobs-unless-theyre-100-qualified

Scikit-learn Documentation,
https://scikit-learn.org/stable/modules/model_evaluation.html

Python for Data Science,
https://jakevdp.github.io/PythonDataScienceHandbook/

Stanford Social Innovation Review,
https://ssir.org/articles/entry/when_good_algorithms_go_sexist_why_and_how_to_advance_ai_gender_equity

Gender Bias In Predictive Algorithms: How Applied AI Research Can Help Us Build A More Equitable Future,
https://www.forbes.com/sites/cognitiveworld/2020/05/30/gender-bias-in-predictive-algorithms/?sh=22de42f957ac

How Artificial Intelligence Can Perpetuate Gender Imbalance,
https://www.oliverwyman.com/our-expertise/insights/2020/mar/gender-bias-in-artificial-intelligence.html