Maxim Sergeev (2696361)

Daniel Siormanolakis (2781196)

**Extra Group 4: Advisory Report**

1. Introduction

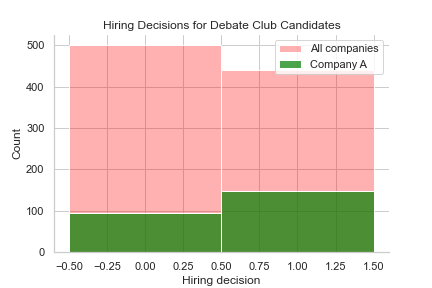
The goal of this report is to improve the hiring process of company A in regards to fairness with our data science analysis conducted on the Utrecht Fairness Recruitment dataset (v. 1.3). Our data science analysis consists of a data exploration approach and three machine learning (ML) models with which we aim to predict whether future candidates would be hired based on past data.

In the following we will explain the ML methods we used for our models, present the results we achieved using these models, draw conclusions from these results and finally make recommendations on how company A could improve their hiring process.

2. Method

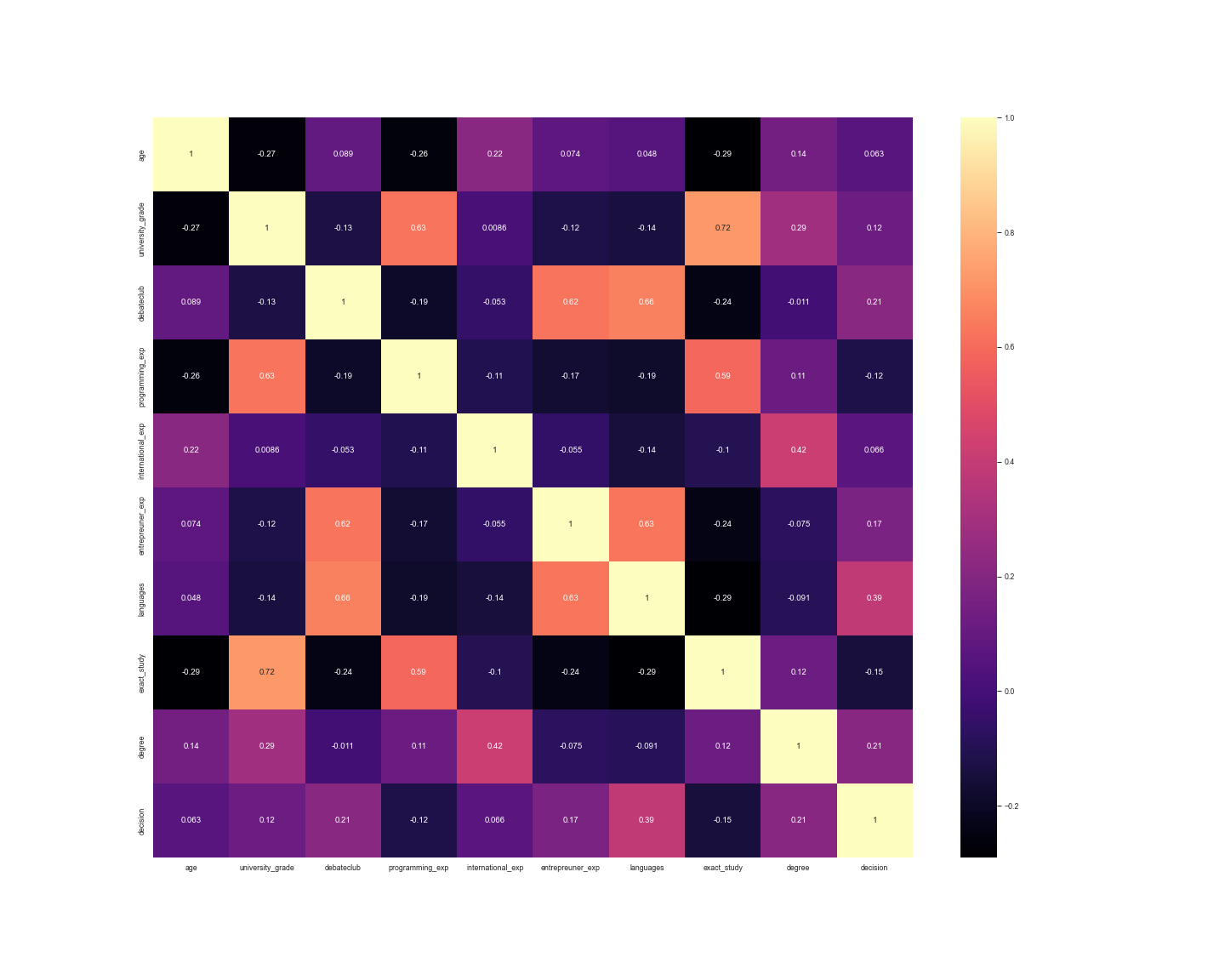
We used two different methods for our three models. Since it was our goal to predict labels (hire/don’t hire) based on previous data, we decided to use supervised learning classifier algorithms. Model 1 (M1) and model 3 (M3) use a random forest decision tree classifier while Model 2 (M2) uses a k-nearest neighbors classifier.

We decided to use only data from company A since our data exploration showed that company A indeed had some idiosyncrasies in their past hiring process. One example for this would be a stronger preference for candidates who mentioned on their CV that they attended a debate club, as can be seen in the following graph:



*2.1 Model 1*

For M1 we used four indicators (correlation can be found in the heatmap below), namely how many languages a candidate spoke, whether they attended a debate club, what their highest degree was and whether they already had any entrepreneurial experience. Our rationale was to just use the four indicators with the highest correlation to the decision.



To train M1 we split our data for company A (n = 1,000) into a training set (n = 800) and a test set (n = 200). We used grid search cross-validation to maximize our hyperparameters (number of decision trees and maximum depth) for the training set. The optimized hyperparameters for M1 are 100 decision trees and a maximum depth of 3. For the cross-validation on the training set we used a stratified k-folds approach with five folds. Afterwards we measured accuracy and F1-score of our model on our test set.

*2.2 Model 2*

For M2 we used as indicators the candidate's university grade, whether they already had any programming experience, whether they already had any international experience, how many languages they spoke, whether they studied a “hard science” as well as their highest degree. Our rationale was a trial and error process based on the accuracy result of our grid search (similar to M1). The hyperparameter optimization and validation were conducted similarly to M1. The optimized hyperparameters for M2 are n\_neighbors = 3, weights = ‘distance’ and algorithm = ‘ball\_tree’.

*2.3 Model 3*

For M3 we only used descriptors as indicators. These descriptors were gender, age, nationality and first sport on the CV. The hyperparameter optimization and validation were conducted similarly to M1 and M2. The optimized hyperparameters for M3 are 100 decision trees and a maximum depth of 1.

3. Results

In the following table you can see the F1-score and accuracy for each of our models on training data:

| **Model** | **M1** | **M2** | **M3** |
| --- | --- | --- | --- |
| **Accuracy test set** | 0.73 | 0.79 | 0.61 |
| **Accuracy grid search (five splits)** | 0.71 | 0.78 | 0.57 |
| **F1-Score** | 0.66 | 0.73 | 0 |

Not only did we compare the individual models against each other, we also compared the models amongst themselves with different indicators or with different subgroups.

For M1 we compared the regular model to models with three and five indicators as well male only and female only subgroups. Unfortunately we were not able to conduct more comparisons on other subgroups.

| **M1** | **Three Indicators** | **Regular (Four Indicators)** | **Five Indicators** | **Male Only** | **Female Only** |
| --- | --- | --- | --- | --- | --- |
| **Accuracy** | 0.73 | 0.73 | 0.69 | 0.73 | 0.73 |

On our test M1 had a 39.5 % acceptance rate (77 out of 200). Out of all accepted candidates, there are 67.5 % who are considered false positives (candidates who would usually be rejected). Out of all rejected candidates there are 22.3 % who are considered false negatives (candidates who would have usually been accepted). All false negatives only have a bachelor’s degree.

For M2 we also compared male only and female only subgroups.

| **M2** | **Regular** | **Male Only** | **Female Only** |
| --- | --- | --- | --- |
| **Accuracy** | 0.79 | 0.75 | 0.70 |

We did the same procedure for M3.

| **M3** | **Regular** | **Male Only** | **Female Only** |
| --- | --- | --- | --- |
| **Accuracy** | 0.61 | 0.48 | 0.60 |

4. Conclusions

It is very difficult for us to draw conclusions from our results. If we would have had more capacities (*aka* people in our group…) and time we would have conducted further tests on subgroups.

M3 with an F1-Score of 0 is basically useless. It indicates that it has a recall of 0; it just assigns “don’t hire” to every column. Why it behaves like this is unclear to us.

M2 seems to perform better on males than on females. The exact reason for this is unclear to us. From our data from company A we found out that hiring chances for women are 0.39, while for men they are 0.47. However this does not explain the drastic discrepancy specifically for M2. M2, since it doesn’t use any descriptors, aims for a fairness through unawareness approach which, as shown above, fails.

M1 has a constant accuracy across all tests. Even though it performs slightly worse than M2, it does not seem to be biased against either women or men. Apparently here the fairness through unawareness approach works. However we can not say how fair M1 (or for that matter any of our models) is with regard to other protected attributes such as age or nationality.

5. Recommendations

Based on the conclusions we would recommend company A to incorporate M1 into their hiring process. In the following we want to show what this improved hiring process could look like.

First and foremost it is important to note that M1 will not be able to automatize the hiring process entirely. The reason for this is that every prediction model based on past data will carry some bias. Good prediction models can (ideally) correct for this to some extent but never entirely.

HR should be adamant to check for CVs of candidates who only have a bachelor’s degree! M1 is highly biased against this subgroup.

To improve and recalibrate the model, we recommend that a certain number of CVs of subgroups that are heavily underrepresented (females, other gender, chess players, older applicants) should be reviewed and documented manually.

Company A could put the following text on their website to explain the hiring process to potential applicants:

“To streamline our recruitment process we incorporate a machine learning model into our hiring process. Every applicant’s CV will be scanned to decide whether they will receive an invitation for an in-person. This process has the advantage of being faster than traditional CV scan. Should you receive a rejection for an in-person interview you have the possibility to ask for a second opinion from one of our HR representatives. In case you will get a rejection again, you will be provided with an explanation for why we decided against your CV.”

On average company A uses 100 work hours every month on manual CV checks. Assuming our model operates on an acceptance rate of around 40 % and every rejected candidate asks for a second manual scan, we can reduce this to 60 work hours. Even if you add another 10 hours to recalibrate the model every month, the company would still have the benefit of saving around 30 work hours every month.