# PROJECT

## HR DEPARTMENT AI

# HR CASE:

code academy

01 This is good boy!

02 This is a beautiful girl.

03 This girl is my aunt's daughter.

04 This boy is my brother.

05 This boy is my uncle's son.

# PROJECTS

Our first project is the selection of the most suitable employee by means of a model.

Our second project is a project designed for the candidate to test himself. That is, it checks whether the candidate's CV is suitable for the job.

code academy

# 01

## Vacancy

HR receives an order to give a vacancy. The vacancy is distributed.

## Candidate

Hundreds of candidates apply for the vacancy.

800

## İnterview

Resumes are reviewed and candidates are invited for an interview.

800

## Model

We feed the candidate's data into the model and the model returns us the top 10,100 candidates.

100

## Interview

At the interview stage, we interview 100 people, not 800 people.

## Cost

code academy

# COST REDUCTION

Finding an employee for a vacancy may take some time. We minimize that time by reducing the options.

We are saving financially. First, we waive the cost of calling for each candidate. Second, we don't pay more hr workers.

In conclusion, we find the best candidate that meets our requirements.

code academy

# STATISTICS:

A lower estimate: 1% of 17,000 = 170 HR department employees.

A mid-range estimate: 2% of 17,000 = 340 HR department employees.

An upper estimate: 3% of 17,000 = 510 HR department employees.

code academy

# 02

### Vacancy

HR receives an order to give a vacancy. The vacancy is distributed.

### Candidate

The candidate enters his/her cv in the model section.

### Model

The model reviews the features of the cv and displays the candidate's selection response.

### Suitable

### Unsuitable

code academy

# PROFIT:

Knowing whether the candidate is suitable for the vacancy in advance.

Innovation in the industry. This project is the first in our company.

Operability and Convenience. Get instant and convenient information without showing your CV to anyone.

code academy

# ALGORITHM:

**Unsupervised Learning**

DBScan 01

**Supervised Learning**

Gradient Boosting 02

# CANDIDATE DATA

| received_callback | firstname | race | gender | years_college | college_degree | honors | worked_during_school |
|---|---|---|---|---|---|---|---|
| 0 | Allison | white | f | 4 | 1 | 0 | 0 |
| 0 | Kristen | white | f | 3 | 0 | 0 | 1 |
| 0 | Lakisha | black | f | 4 | 1 | 0 | 1 |
| 0 | Latonya | black | f | 3 | 0 | 0 | 0 |
| 0 | Carrie | white | f | 3 | 0 | 0 | 1 |

| years_experience | computer_skills | special_skills | volunteer | military | employment_holes | has_email_address |
|---|---|---|---|---|---|---|
| 6 | 1 | 0 | 0 | 0 | 1 | 0 |
| 6 | 1 | 0 | 1 | 1 | 0 | 1 |
| 6 | 1 | 0 | 0 | 0 | 0 | 0 |
| 6 | 1 | 1 | 1 | 0 | 1 | 1 |
| 22 | 1 | 0 | 0 | 0 | 0 | 1 |

# FEATURES:

## Year Ratio

Let's create new features and find years ratio (years_college ,years_experience)

## Positive Features

Summarize the columns that positively affect quality ("honors", "computer_skills", "volunteer", and "military".)

## Overall Experience

Calculate cumulative measure of overall experience by summing "years_college" and "years_experience"

## Employment History Summary

Create new feature that counts the number of potential negative attributes like "worked_during_school" (which might indicate a lack of focus on studies) & "employment_holes" (which might indicate gaps in employment history).

# FEATURES:

### Combined Skills

Sum or aggregate "computer_skills" and "special_skills" to create a single feature that measures a candidate's overall skills.

### Avg_mil_work

Worked_during_school + military / 2 (worked_during_school Before Corr: 0.32 || military | Before Corr: 0.31 |After Corr: 0.40)

### Mltp_experspecskill

Years_experience * special_skills (years_experience | Before Corr: 0.08 || special_skills | Before Corr: 0.06 | After Corr: 0.14)

### Substract_comp emp_hole

Computer_skills - employment_holes (computer_skills | Before Corr: 0.23 || employment_holes | Before Corr: -0.21 | After Corr: 0.30)

# SCORE:

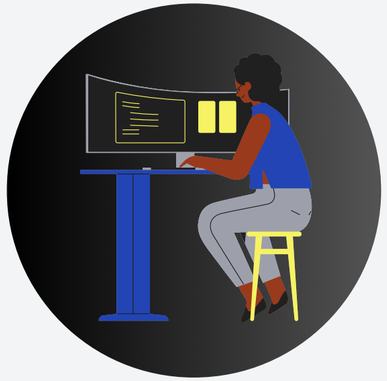
## GRADIENT BOOSTING

I get an f-score of 89% on 'gradient boosting'. I use random search hyperparameters and get this score.

**89%**

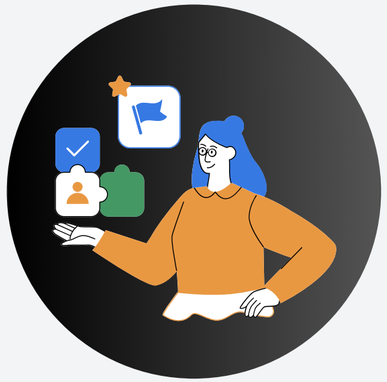
code academy

# QUESTIONS?



## DEVELOPER

The model reviews the features of the cv and displays the candidate's selection response.



## HR

The candidate enters his/her cv in the model section.

**01**

**Add new features**

**02**

**Add weights**

# THANKS FOR ATTENTION!

Presented By Riyad Ahmadov