Hush Hush Recruiter

BORKAR, AADISHREE

SHARMA, SOUMYA

KIONGERA, MERCY

HANIF, ABDULLAH

Agenda

01 Project Architecture

02 Data Sources & Cleaning

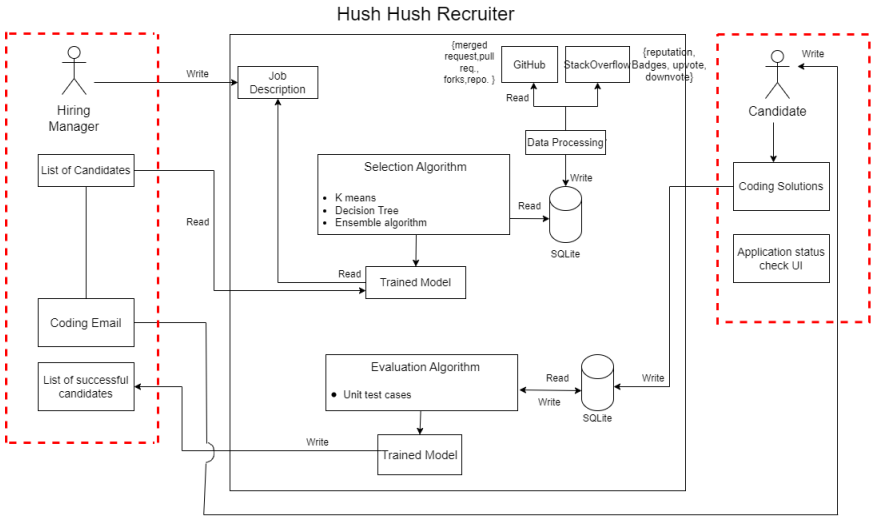
03 Database ER

04 Selection Algorithm Overview

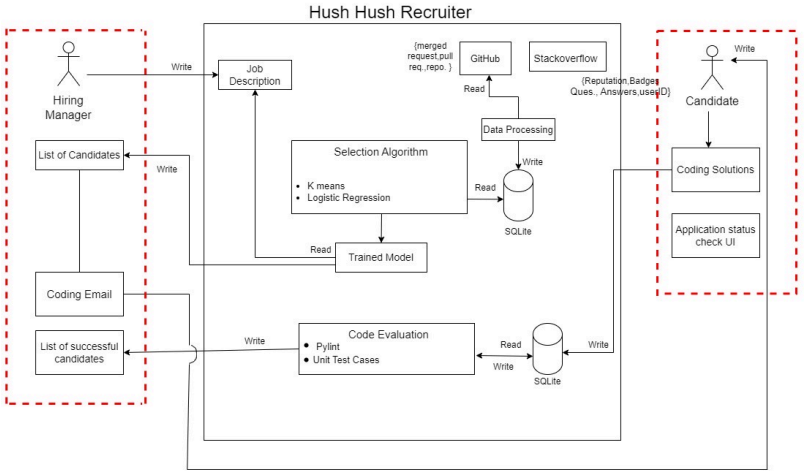
05 Code Evaluation Overview

06 Demo

Project Architecture

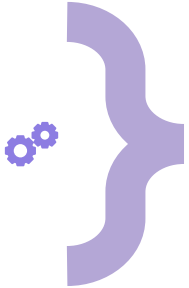


What we promised



What we achieved

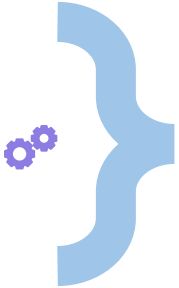
Data Sources, Technologies & Cleaning



Data Sources

Stack Overflow

Github

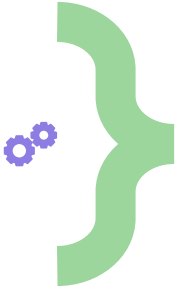


Technologies Used

Async IO, Requests

PyGithub, AIOHTTP, Time

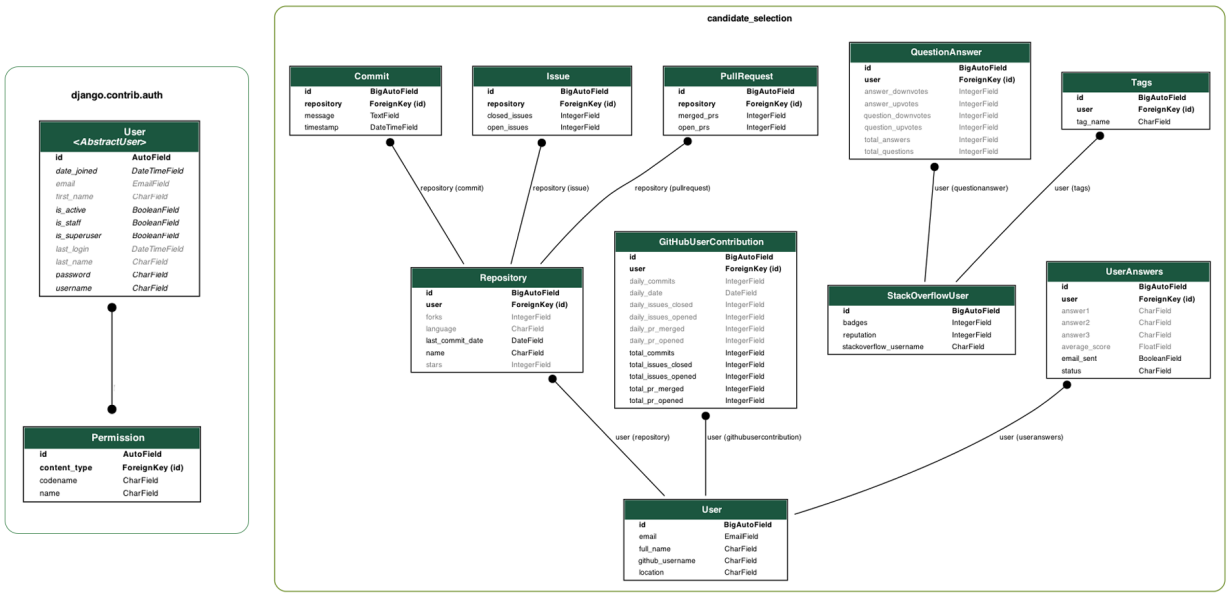
Stack exchange API



Data Cleaning

Removed null values for required fields.
Converting data types
Replacing Null values with 0

ER DIAGRAM


django.contrib.auth
User
<AbstractUser>
id AutoField
date_joined DateTimeField
email EmailField
first_name CharField
is_active BooleanField
is_staff BooleanField
is_superuser BooleanField
last_login DateTimeField
last_name CharField
password CharField
username CharField
Permission
id AutoField
content_type ForeignKey (id)
codename CharField
name CharField
candidate_selection
Commit
id BigAutoField
repository ForeignKey (id)
message TextField
timestamp DateTimeField
Issue
id BigAutoField
repository ForeignKey (id)
closed_issues IntegerField
open_issues IntegerField
PullRequest
id BigAutoField
repository ForeignKey (id)
merged_prs IntegerField
open_prs IntegerField
QuestionAnswer
id BigAutoField
user ForeignKey (id)
answer_downvotes IntegerField
answer_upvotes IntegerField
question_downvotes IntegerField
question_upvotes IntegerField
total_answers IntegerField
total_questions IntegerField
Tags
id BigAutoField
user ForeignKey (id)
tag_name CharField
repository (commit)
repository (issue)
repository (pullrequest)
user (questionanswer)
user (tags)
Repository
id BigAutoField
user ForeignKey (id)
forks IntegerField
language CharField
last_commit_date DateField
name CharField
stars IntegerField
GitHubUserContribution
id BigAutoField
user ForeignKey (id)
daily_commits IntegerField
daily_date DateField
daily_issues_closed IntegerField
daily_issues_opened IntegerField
daily_pr_merged IntegerField
daily_pr_opened IntegerField
total_commits IntegerField
total_issues_closed IntegerField
total_issues_opened IntegerField
total_pr_merged IntegerField
total_pr_opened IntegerField
StackOverflowUser
id BigAutoField
badges IntegerField
reputation IntegerField
stackoverflow_username CharField
UserAnswers
id BigAutoField
user ForeignKey (id)
answer1 CharField
answer2 CharField
answer3 CharField
average_score FloatField
email_sent BooleanField
status CharField
user (repository)
user (githubusercontribution)
user (useranswers)
User
id BigAutoField
email EmailField
full_name CharField
github_username CharField
location CharField

Candidate Selection Algorithm

- Input data

Github user dataframe , 5 datapoints (commits, opened pull requests, merged pull requests, issues opened, issues closed)

- ML library used

WHY?

- ✓ Simple & efficient.
- ✓ Includes wide range of algorithms.
- ✓ Integrates seamlessly with computing libraries such as Numpy.
- ✓ Open source.

*Images taken from Google.

Candidate Selection Algorithm

- ML algorithms implemented

K means clustering -
- ✓ Defined number of clusters as 2 (1 -good , 0 –bad).
- ✓ Trained model using kmeans.fit()
- ✓ Used Pickle library to serialize the trained model into a .pkl file.

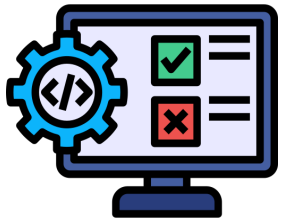
Logistic regression-
- ✓ Taking the user dataframe along with cluster labels and validating the good & bad candidates.
- ✓ Trained model using logreg.fit()
- ✓ Used Pickle library to serialize the trained model into a .pkl file.
- ✓ Printed accuracy and confusion matrix (Accuracy was 0.96).

Image source-https://www.interviewquery.com/p/data-science-memes

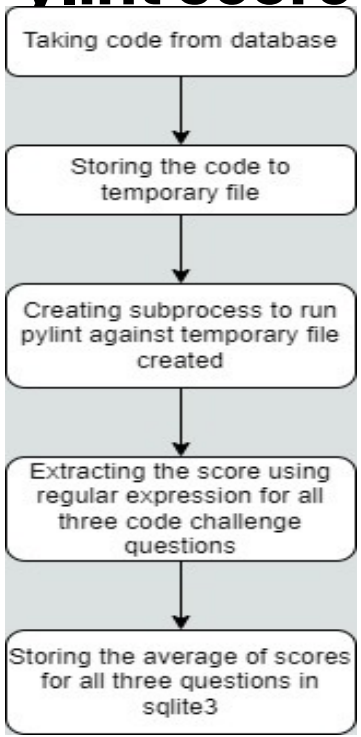
Candidate Selection Algorithm

- Testing the model

  ✓ Depending on user requirement, new data is scaled & passed through model.
  ✓ Both models are loaded using pickle.load
  ✓ Output list of candidates is seen on the UI.

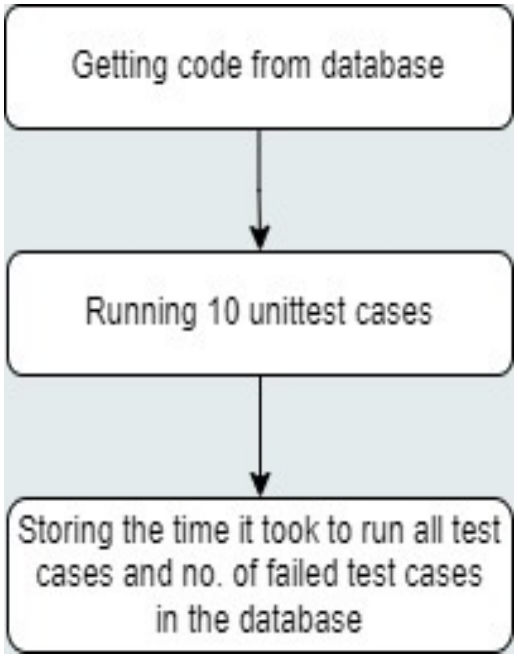


Code Evaluation: Pylint Score



- Libraries used:
  a) Pylint
  b) re
  c) Subprocess
- We are using Pylint to check code and its quality.
- We are using 're' package to extract the evaluation score from the output report.
- We are running pylint in the subprocess because we can isolate pylint issue with main python code.
- Refrence: https://pylint.pycqa.org/en/latest/development_guide/api/pylint.html

Code Evaluation: Unittest Cases



- Libraries used:
  a) Unittest
  b) re
  c) sys
  d) io

- The primary motive behind using unittest is to verify the behavior and correctness of individual units of code.

- It is also implemented to give us idea about the time complexity of the code submitted by the applicant.

Email Executions

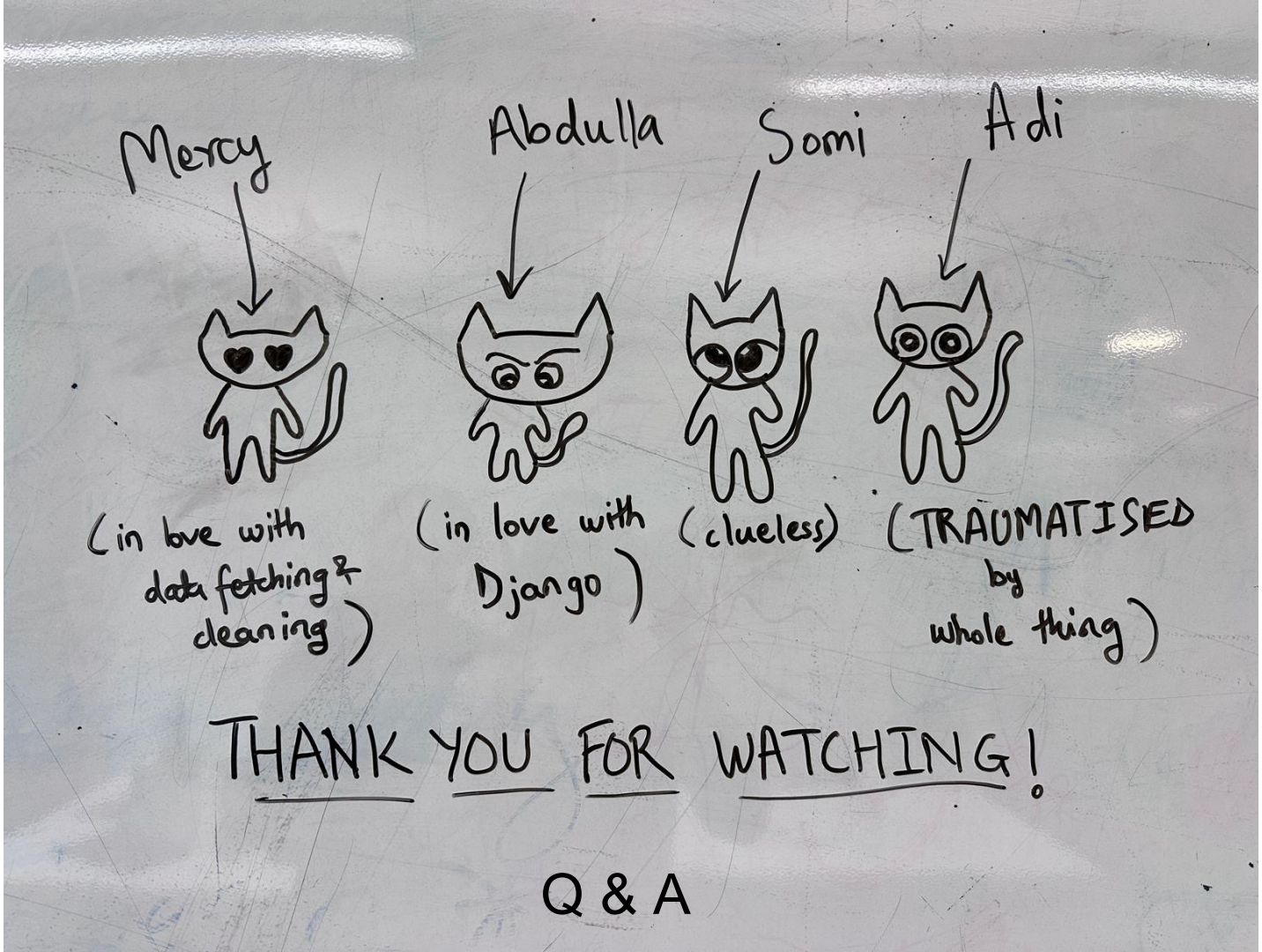
- Email to invite selected candidates to an interview
- Email to rejected candidates
- Email to invite selected candidates for an interview

Technology Used:

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 465
EMAIL_USE_SSL = True

EMAIL_HOST_USER = os.getenv('EMAIL_HOST_USER')
EMAIL_HOST_PASSWORD = os.getenv('EMAIL_HOST_PASSWORD')
DEFAULT_FROM_EMAIL = 'engabdullahhanif@gmail.com'
```

Live Demo


Mercy
Abdulla
Somi
Adi
(in love with data fetching & cleaning)
(in love with Django)
(clueless)
(TRAUMATISED by whole thing)
THANK YOU FOR WATCHING!
Q & A