|  |  |  |  |
| --- | --- | --- | --- |
| Category Max | Mark | Team | Notes |
| Technical Depth and Novelty (45%) | 9 |  |  |
| Implementation far from completion | 2.3 |  |  |
| Complete implementation according to the scope of the categories to the left project objectives without solving technical challenges | 4.5 |  |  |
| Complete implementation and solving some technical  challenges | 6 |  |  |
| Completed with some degree of technical novelty | 7.5 |  |  |
| Completed, with good degree of technical novelty, and novelty in addressing project objectives | 9 |  |  |
| Correctness and Performance (30%) | 6 |  |  |
| Unacceptable performance, buggy even with a few tests | 1.5 |  |  |
| Overall correct but slow | 3 |  |  |
| Overall correct and efficient | 4 |  |  |
| No issues during demo and project testing (by the assessors) | 5 |  |  |
| Robust and excellent performance | 6 |  |  |
| Code Style, Structure, and Readability (12.5%) | 2.5 |  |  |
| Messy code structures, difficult to read | 0.7 |  |  |
| Readable but not organized | 1.3 |  |  |
| Code is well structured and readable with some documentation | 1.7 |  |  |
| Well structured and easy to read with ample documentation | 2.1 |  |  |
| Easy to read, well documented, and demonstration of excellent coding style and practice | 2.5 |  |  |
| Interface and Usability (12.5%) | 2.5 |  |  |
| Primitive interface and difficult to use | 0.7 |  |  |
| Poor interface design but still usable | 1.3 |  |  |
| Generally good design with usability issues on some use cases | 1.7 |  |  |
| Generally good design and ease to use in all aspects | 2.1 |  |  |
| Professional interface design and excellent usability | 2.5 |  |  |
| Total Mark (out of 20) | 20 |  |  |

15-20 mins

functionality

novlety

tell a story

discuss engineering practice, methodology

* agile pratice
* master->indivudial branch
* .ignore
* snippets
* machine learning for user-similarity
* automatically switch between development and online environment
* integration of the system: web host <-> Google SQL cloud
* logic layer, business layer, ..
* library, API, service
* Django, Google SQL cloud, SMTG mail service

link function -> story, high level region?

* implementation issue: Sklearn with apache2
* how to deploy
* how to use
* easy to read