

GSoC Project Proposal

Organisation : Digital Impact Alliance

Sub-Organisation : SUMSarizer

Student's Information

Name and Contact Information

Name : Surya Saini

Email : sainisurya1@gmail.com

Country Code : +91-9729663866

Time Zone : UTC + 5:30 (IST)

Blog URL : [God A Myth or Reality](#)

Github : [Dragneel7](#)

Resume : [Link](#)

Education

University : Indian Institute of Technology, Roorkee

Major : Electronics and Communication (Batch of 2020)

Current year : Second

Degree : Bachelors in Technology (B.Tech)

Additional

Homepage : [Homepage](#)

HOTOSM - Slack Handle : Surya Saini

Twitter : [Surya Saini](#)

LinkedIn : [LinkedIn profile](#)

About Me

I am a sophomore, currently enrolled in Electronics and Communication engineering at IIT Roorkee. I developed a passion for programming, web development and machine learning in my freshmen year. I have been contributing to open source regularly since about six months now - looking over repositories of the products I use/like, trying to give back my part of contribution back to the organisation.

I have an experience of working closely with a team as I am an active member of **Information Management Group** at IIT Roorkee, a bunch of passionate enthusiasts who manage the **institute main website**, internet and intranet activities of the institute.

Coding Skills

I am comfortable with all the technologies used in the development of this project. I have a strong background in web development, with proficient knowledge in version control system (like git), MTV and MVC architectures. I can write (in order of proficiency)

- ❖ Python : programming in python, Flask web framework, Django framework, Django Rest framework
- ❖ Machine Learning : Python(proficient with scikit learn), R
- ❖ Javascript : JQuery, Vanilla JS, AngularJS, Angular 5
- ❖ Design : Adobe Illustrator, Adobe photoshop

Project

PredictX

Build as part of OpenAI Hackathon, the app was developed to help the students predict their scores in the examination based upon their previous scores and other factors that may affect their grades. The app also provided them with suggestions on how to alter these grade affecting factors to improves their marks and predicted the colleges they would get based on their z-score.

Save The Clash

Using state of the art object detection, segmentation and monocular depth estimation networks, developed an app to help the visually impaired navigate in an unknown environment. Providing them with a sense of digital vision.

Rtograph

Build with an aim to help the students of the campus, the app brings all the information of the university and graphically displays it on a map overlay. It helps students to search for almost any information with a single query. Rtograph can be aptly called with graphical wiki of IIT Roorkee.

The app is build with Django Rest as backend and has a frontend developed on Angular 5 framework. A lot of challenges were solved in the development of the app. Handling a large data and displaying it on the map, making it look good and having a great UX at the same time.

Graph CNN

Extending image classification to non-euclidean domains using graph based CNNs

Pre GSoC Involvements

SUMSarizer:

#40(issue)	Ability to see/edit tags
------------	--------------------------

#51(issue)	Error while building the app in deployment mode
----------------------------	---

Apart from SUMSarizer, I have been regularly contributing to other open source organisations too.

HOTOSM/Field-Campaigner:

#416(merged)	Fixed typographical errors in README
#419(mergeable)	Improved UX for campaign view
#420(mergeable)	Add real time user location on map
#417(issue)	Deploy the application in development mode.

Fossasia:

#310(merged)	Added exception handling for "Ask.com" web-scrape
------------------------------	---

Aima-javascript:

#133(mergeable)	Add performance measure for simple logical agents
#138(merged)	Fix issue link for A* search
#136(mergeable)	Add Min-Conflict Visualization
#153(mergeable)	Add visualization for Expectiminimax search

Aima-python :

#665(merged)	Fixed typographical errors in agents.py and agents.ipynb
#669(merged)	Updated README
#685(merged)	Added heuristics for A* function
#689(merged)	Solved issue of loading search.ipynb
#697(mergeable)	Added example for hill climb search

Aima-exercises:

#6(merged)	Added file lookup-table in README
----------------------------	-----------------------------------

Aima-glossary:

#4(merged)	Added glossary terms
----------------------------	----------------------

Aima-data:

#13(mergeable)	Added quadragram file for hill climb search
--------------------------------	---

Project

Implement Super Learner 3 in SUMSarizer

Motivation

Implementing Super Learner 3 to SUMSarizer, replacing the current Super Learner algorithms in the feature selection and classification files. Present implementation establishes learning as a 2 stage process. Stage 1 : establishing the probability of an individual temperature observation and Stage 2 : aggregating the readings obtained from stage 1. The aim for the project would be to replace the algorithms with those of Super Learner 3, enabling this multi stage learning explicitly.

My motivation for GSoC this year is to get started with open source. GSoC is a great program for introducing organisations with prospective contributors. The main reason I wish to work with SUMSarizer is to provide my share of contributions towards the project and learn and develop my skills of machine learning at the same time, bringing them to good use.

Deliverables

During GSoC my main task would be to port the machine learning algorithms from current Super Learner to Super Learner 3. My goals for achieving this would be :

- **Reimplementing features.R** : This part would work in collaboration with the student working on the other proposed project “Adding data importing interface to ingest data from other common SUMS models and arbitrary time series data”. The new implementation will extract features from different type of file formats and more classification features will be incorporated(depending on user input data). These features will use the inbuilt feature extractions libraries in R.
- **Implement Super Learner 3 in ml_learner.R** : Replace the algorithms working on origami/SuperLearner with Super Learner 3. It would require the understanding of the library, enabling a more efficient way of stage learning.
- **Modifying ml_worker.py** : modifying the main python file that runs the entire classification to work with the changes implemented.

The above proposed tasks would achieve the goal of GSoC task. I plan on solving some pre-existing issue and add new features for the development of SUMSarizer as whole.

- **Select algorithm in Stack Regression** : The current implementation in the file ml_learner.R has preselected machine learning algorithms in a stack that are used for classification task. It would be more useful to allow the user to select the algorithms he thinks are the best for the data they provides. For doing so we can suggest them the type of algorithms that fit better on different portions of data(basics of stack regression). This would require a additional view/template where the user will choose the algorithms and a modification in ml_worker.py and ml_learner.R to incorporate these additional user parameter.
- **Ability to see/edit another labelers label [#40](#)** : Adding functionality similar to google docs where an invite can be sent to another labeler, accepting the invitation would allow certain permissions over another labelers labels. These permission could be view, edit or comment suggestion.
- **Non Study owner should be able to SUMSarize [#39](#)** : This issue would require changes in the model structure and views, allowing permissions to non-logged in users to view study.
- **Prompt User to add some information about what they are using SUMSarizer for [#34](#)** : “Data is power”. Adding another model in DB to store this information would help in development of SUMSarizer. Getting to know what different users use it for we can provide them with individual suggestions and improve their SUMSarize experience. It as a whole is a huge project, I would focus on the UI/UX component for asking this information and storing it in the database.

Proposed Timeline

Community Bonding Period(23 April - 14 May)

I will utilise this time to get more familiarized with flask, R, read the documentation, discuss upon the techstack to be used and get to know my mentor and other fellow community members so that we can work together with an amplified efficiency. Most importantly I would discuss upon the [issue #32](#) i.e. replacing R with python. This decision would influence major portion on my proposed GSoC task.

Week 1-2 (May 14th - May 27th)

Discuss the permissions and Implement view/edit label feature. Would require to change the views add additional views and change in DB structure to provide sudo-user authorities to the requested labeler.

Week 3 (May 28 - June 3)

Read and implement mock project on R or Python(based in the community period decision) using the libraries to be used in SUMSarizer to get myself more acquainted with the functioning and decision making of the application.

Week 4 (June 4 - June 10)

Implement the feature.R file incorporating the features mentioned above. The deadline for this may vary with the work of my fellow GSoC student working on the other 2 projects.

Phase 1 evaluation submission**Week 5-6 (June 11 - June 27)**

Implement view for selecting the algorithms to be used for stack regression, modify the ml_worker.py file and start implementing Super Learner 3 in ml_learner.R file.

Week 7-8 (June 28 - July 15)

Finish implementing Super Learner 3 in SUMSarizer and start working on issue [#39](#).

Phase 2 evaluation submission**Week 9-10 (July 16 - August 1)**

Work the take user information feature. Submit design mockups for the same and implement it.

Week 11-12 (August 2 - August 14)

Solve issue [#41](#) and make a video in collaboration with my fellow GSoC colleague to help users understand the workability of the new SUMSarizer(new views and features added).

Week 13 (Conclusion) :

Create documentation for usage with Super Learner 3 and SUMSarizer and add blog for GSoC experience.

I have applied to HOTOSM and AIMACode for GSoC but if selected for SUMSarizer, I will work to implement the machine learning algorithms, learning and contributing together.

I believe I have enough fuel to get started on my goals, I plan to devote my entire summer for the completion of the project

I do not have any other commitment during GSoC period. During this time period I'll be able to devote 40-50 hours a week till my college reopens and 30-35 hours per week after that. I am free on weekends.

I have my exams from 26 April - 4 May (Community bonding period). I will utilize this time to get to know my mentor and decide the workflow to be followed. My university will reopen around Week 9 i.e. July 15, from then I will try to devote 30-35 hours per week. I will be free on weekends I will devote more time towards completion of the project..