**Weekly Progress**

1 August – 7 August:

**Mentor Selection and Topic Discussion**

* We formed our team and chose a mentor for us.
* We pitched and had discussions about various ideas for the project.

8 August – 14 August:

**Topic finalization**

* We discussed with our mentor that how to approach the project.
* We decided to conduct weekly meetings to assess the progress.
* We started to learn the basics of NLP and other topics related to project.

15 August – 21 August:

**Field Segregation**

* We decided to use FReMP stack for the project prototyping.
* We divided ML and Flask part among team members to learn.
* We were still learning NLP from online educators and exploring Kaggle for available data sets related to the project.

22 August – 28 August:

**Dataset Exploration and Learning Concepts**

* We finalized our dataset- **CMU Book Summary Dataset**. This dataset contains plot summaries for 16,559 books extracted from Wikipedia, along with aligned metadata from Freebase, including book author, title, and genre.
* We started exploring the dataset and looked at key features in the data.
* Most of the techniques we studied under **NLP** were applicable to single-label classification but books generally don’t have one theme. This proposed a challenge.
* We found out some resources on analytics Vidhya on **Multi-Label Classification** and started learning these concepts to make basic ml models.
* Along with learning Multi-Label Classification, we decided to start preprocessing and cleaning the dataset.
* We removed all the genres having a very low frequency, cleaned the dataset and removed stop-words.
* We also plotted various graphs to analyze the dataset better.

29 August – 4 September:

**Using Basic ML models**

* Now that we’ve learned basics of **Topic Modelling** and **Multi-Label Classification**, we decided to implement these techniques on our dataset.
* We used *sklearn’s* ***MultiLabelBinarizer* to** one hot encode the target variable i.e genres and used concepts of *tfidf, word2vec and count vectorization* to extract features from the cleaned version of movie plots.
* We started training simple ml models like **logistic regression** and **NB** to classify the genres and decided to go along with **logistic regression**.
* We achieved quite a low accuracy as we didn’t tune in the hyperparameters.
* We tuned in parameters during vectorization of features like *max\_df, min\_df, ngram range,* etc and to analyze their effect on accuracy, we plotted various graphs and justified the effects.

5 September – 11 September:

**Finalizing a model**

* After advice from our mentor, instead of dividing the dataset into **training** and **testing**, we divided it into **training**, **validation** and **testing** so that the final testing data provides an unbiased evaluation.
* We explored all possible ways of achieving **multi label classification** using **supervised learning**.
* We were using *sk-learn’s* OneVsRestClassifier class to solve this problem as a **Binary Relevance** or **One-vs-All problem**.
* Now that we already had tuned in hyperparameters during vectorization, we decided the tune in hyperparameters of **logistic regression**.
* We tuned in parameters like *C, solver, penalty* etc and to analyze their effect on accuracy, we again plotted various graphs and justified the effects.
* We tried to vary the thresholds to get the accurate threshold. Decreasing the threshold to around 0.25 proved quite beneficial as it gave maxima for accuracy. This was our final optimization.
* After using multiple optimizations, we achieved an accuracy of around 55% and decided to finalize this model further in our project.

12 September – 18 September:

**Deployment and Change of Technologies:**

* After creating the model, we used flask to create a basic prediction website hosted at “<https://bookgenics.herokuapp.com>.”
* To make things more scalable and more comfortable to manage, we started to learn the MERN stack.
* The parts of the MERN stack to be learned were distributed among the group members.

19 September – 25 September:

**Still in the learning phase**

* At this time, we were still learning the MERN stack.
* We were making some small-scale apps with the stack to understand its core concepts.

26 September – 2 October:

**First MERN prototype**

* After we had learnt the technologies required, we started to make the web part of the project again.
* We built the first prototype of the project which just had 4 pages – home page, login page, register page, and dashboard page.
* After registering, the user was required to login and then he/she could upload the story and get the predicted genres.
* The user could log out also. This was the most simplistic version of website that we built.
* We built the basic backend config using node.
* We used cloud mongo storage and also local mongo storage for the database of books.
* We made routes and middle-wares for authenticating, predicting genres and registering the user in the database.

3 October – 9 October:

**Refactoring whole project and some major changes**

* We started with changing the home page, we changed the background and did other CSS changes.
* We refactored the login and register page, we introduced more styling, images, used better bootstrap, also user could now receive errors while registering and logging below the fields itself.
* While registering we also made a page where user would have to select his favorites genres among the list of 15 genres provided.
* We made the website mobile responsive.
* After login we made a total of 5-page options – home dashboard, predictor page, books added page, favorites page, and settings page. All of them were not functional right now.
* We made a working dashboard navbar, logout button was also used again.
* We made simple backend routes for get the books, and other things related to books and genres.
* All the books were fetched on dashboard and displayed to the user.

10 October –16 October:

**More features were added**

* We added the sorting books and filtering books feature.
* We reused the code and made the books added and favorites page functional.
* We created the backend code for getting the favorites of a person.
* We added heart icon on every book which on clicking added it to db.
* The user that has added the book or is admin can edit the book.
* Editing/ Deleting the book feature was also added.

17 October – 23 October:

**Final Changes**

* Reading the content of book feature was added to the books.
* Predictor page was finalized, and books could now be added to the database.
* Books can be added to the database only after predicting its genres.
* Settings page was made with password change feature.
* Python code was integrated with Js code in the backend.
* Some other major changes were made to make the website fully functional.
* Website was hosted at [powerful-lake-91115.herokuapp.com](https://powerful-lake-91115.herokuapp.com).