Sprint No. 1: Improve the personalization of human-generated essays using LSTM models with dropout to minimize overfitting

NOTE: Tasks below can still be modified depending on the requests of the assigned Customer/ Product Owner

1. Backlog Creation and Prioritization

Steps	Task	User Story	Story Points	Sprint No. / Week No
Data Collection	Collect a diverse set of human-generated essays.	Collect human-generated essay that are joy, happiness or excitement focused	2	SW 1-1
		Collect human-generated essay that are sadness or grief focused	2	
		Collect human-generated essay that are fear focused	2	
		Collect human-generated essay that are historical or culturally inclined	2	
		Collect human-generated essay that discusses personal opinions on topics (opinionated essays)	2	
		Collect human-generated essay that are comprising of slang words or idiomatic expressions	2	
Data Preprocessing	Clean and preprocess the data to remove noise and standardize formats.	Clean and preprocess the data to remove noise and standardize formats	1	SW 1-1

Model Design	LSTM models incorporating dropout layers	Experiment on incorporating different concepts to avoid model from memorizing (i.e. dropout, etc.)	3	SW 1-2
Model Training and Validation	Train models using the preprocessed data and validate their performance.	Train models using the preprocessed data and validate their performance.	3	SW 1-2
Evaluation	Evaluate model performance in terms of personalization and overfitting. Iteration: Based on feedback, iterate on the model design and training process.	Evaluate model performance in terms of personalization and overfitting. Iteration: Based on feedback, iterate on the model design and training process.	3	SW 1-2

2. Tools and Technologies

Data Preprocessing: Python (Pandas, NumPy), Jupyter Notebook.
Model Development: TensorFlow or PyTorch for building LSTM models.
Version Control: GitHub for source code management.
Project Management: Jira or Trello for tracking tasks and progress, Teams or WhatsApp for team's communication, Tableau for burndown chart

3. Roles and Technologies

Data Scientist (Roan): Focus on data preprocessing, model design, and evaluation

- Performs the SW 1-1 tasks and updates the User Story in JIRA religiously
- Setups the CI/CD Pipeline in GitHub
- Performs Review of the source code of the Machine Learning Engineer (Peer Review) prior pushing to GitHub
- Performs QA to the output/ user stories implemented of the Machine Learning Engineer

Machine Learning Engineer (Vincent): Implement, train, and iterate on the models

- Performs the SW 1-2 tasks and updates the User Story in JIRA religiously
- Performs Review of the source code of the Data Scientist (Peer Review)
- Performs QA to the output/ user stories implemented of the Data Scientist

Project Manager/Scrum Master (Rashika): Facilitates the sprint planning, daily stand-ups, and retrospectives

- Discusses the Sprint 1 backlogs with the scrum Team if doable for 2 weeks
- Performs the Sprint 1 planning (User Story discussions and Story Points assignment) with the Scrum Team
- Uploads user stories in JIRA
- Creates the burndown chart
- Tracks the progress of the Sprint 1 Activities through JIRA and burndown chart
- Performs daily scrum with the scrum team to discuss – (a) What has been accomplished (b) What is currently being implemented (c) Bottlenecks or issues

Customer/Product Owner (Priyanka): Represents the stakeholders' interests and prioritizes the backlog

- Comes up with backlog or tasks Sprints 1, 2, 3
- Discusses the Sprint 1 backlogs with the Project Manager/ Scrum Master
- Checks the output/ implemented User Stories of the Scrum Team after 2 weeks of Sprint 1(either Accept or Reject)
- Discusses the Sprint 2 backlogs with the Project Manager/ Scrum Master

4. Sprint Planning Meeting

Conduct a sprint planning meeting with the entire team to perform the following:

- Review the backlog and ensure understanding of tasks.
- Estimate the effort for each task using story points or hours.
- Commit to tasks that can be realistically completed in the sprint duration (typically 2-weeks).

5. Agile Practices

Implement Agile practices throughout the sprint:

Daily Stand-Ups: Quick daily meetings to update on progress and identify blockers.
Pair Programming: Encourage collaboration and knowledge sharing, particularly in complex tasks like model design.
Continuous Integration and Deployment (CI/CD): Automatically test and deploy model updates to facilitate continuous improvement.
Retrospectives: At the end of each sprint, review what went well, what didn't, and how processes can be improved.

6. Sprint Review

At the end of the sprint, hold a sprint review meeting to perform the following:

- Demonstrate the achieved work to stakeholders.
- Gather feedback on the model's performance and personalization.
- Adjust the project backlog based on feedback and insights.