- What features did you consider?
  The features considered for the recommendation model are ‘personId’, ‘contentId’, ‘eventType’, ‘text’, ‘title’, ‘timestamp’. Deep Learning recommender uses ‘personId’, ‘contentId’, ‘eventType’. Content-based filtering model uses ‘text’, ‘title’,‘eventType’.

- What model did you use and why?
  I tried a lot of different models including Popularity model, Content-based model(CB), Collaborative Filtering model (CF), Hybrid model (CF and CB), simple Neural Network and Deep Learning models.
  Started with the Popularity model to get a baseline model. Then, tried the collaborative filtering model which gave better results than popularity and content-based models.
  Next, I tried the hybrid model of CF and CB which gave better results than any individual models.
  Next, I tried the deep learning model which gave the best results out of all the models.

- What was your evaluation metric for this?
  Recall@5 and Recall@10 were used as the evaluation metrics to determine the performance of the recommender system model.

  Achieved a perfect score of 100% for Recall@10 for the deep learning model. This means 100% of interacted articles in test set were ranked by the deep learning model among the top-10(from randomly chosen 100 articles)

- What features would you like to add to the model if you had more time?
  Features such as ‘userRegion’, ‘userCountry’, ‘sessionId’ can be added to provide contextual information to the model. These features can enhance the performance of the recommender model.

- What other things would you want to try before deploying this model in production?
  Write clean, readable code. Handle error handling and exceptions cases. Test the model well for different types of inputs. Use distributed computing to handle large scale real-world datasets.