* **Your approach used and what you liked and disliked about this challenge**

The approach I used was vector space modelling using the Doc2Vec module provided by genism. I divided the task into 4 main steps – understanding the data, cleaning and making necessary modules ready for further steps, model building and training and finally testing. Once I had understood the data format and its small details, I prepared a few mappings of the clip ids with their categories, the category names, the titles, images and captions.

Next, I cleaned the data by removing the stop words, punctuation marks. Further, I tokenized the sentences into words using a word tokenizer module provided by NLTK. Next, each sentence (combination of words captured from the title, captions and category names of the clip) had to be tagged so as to identify it in the vector space. For this I used the gensim module for tagging each such sentence with the clip id.

Next task was to vectorize these sentences. I played with the vectorization parameters and trained the model to give accurate similar suggestions. Finally, I tested the model by providing various clip\_ids to the saved model and checked the similarity of the results with the given clip details.

I loved a lot many things of this challenge. Firstly, the formats in which data was provided required me to completely understand the data. Initially I went on to directly start coding but soon realized the future pitfalls, which was fun! For some videos, the data did not have any categories listed, which made the task challenging from a testing perspective where the similar videos had to be extracted only from the context of the words appearing in the titles, captions of the clip\_id. One thing that was confusing was in the given data, for example in the data file "similar-staff-picks-challenge-clips.csv”, there were two columns for the clip\_id. For most of the rows, the values in these two columns were same. However, for one row they mismatched which disturbed all the rows below it. This made it a bit confusing as to which clip\_id’s data is this exactly. Apart from that, it was a lot fun to play with the data. The more I digged into it, the more details I got to know which was very interesting.

* **How would you go about building this for real at Vimeo? Are there additional techniques you would leverage to improve the results?**

Given a chance, I would apply more data cleaning processes before vectorising the words. For example, in the captions/descriptions column of the data, there were many videos for which there was information about the people involved in making the film. Or there were email addresses for more details. I feel these details were not necessary. For this, multiple iterations would be required to find the common keywords that appear in all the captions in general and then remove the unnecessary words. Next, I would use some computer vision techniques like object recognition from the given images to leverage the given image data as well. Objects could be classified into specific classes which can be added as words to the vectorization sentence.

Other technique that would improve the results is to continuously update the data file "similar-staff-picks-challenge-clip-categories.csv” in every iteration. This is because, currently not all clip-ids are listed in this file with their respective categories. So, now when we have the most similar categories for a clip\_id, we can aggregate over all the categories to which the 10 similar videos belong and label this input clip-id with the common category. This can be updated in every iteration of model training.

Next, while implementing on a platform like Vimeo, it is not only important to suggest videos that are similar to the current video, but also to look at who is watching the video. Meaning, the current user’s previous preferences should also be taken into consideration while suggesting videos. These suggestions could be completely different from the video currently being watched, but it may interest the user. Then considering preferences of people with similar tastes as to this user could be used to make suggestions. Building such a model would be an interesting task!

* **What performance considerations are important to scale this up to all Vimeo videos and why?**

The time and space considerations are important given that there will be much more data than given in this challenge. This means, the model training and storage has to be optimum. The feature set being huge, the model building phase would require a higher number of worker threads. For this, auto-scaling options like using cloud resources could be one solution. As new videos are uploaded every minute, users would have to be suggested with latest, popular videos. This means there has to be some priority-enforcing module. Next, considering the load on the servers, load balancing is also an important performance consideration. Data and services availability should also be considered. For this, server duplication is required. Finally, data privacy and security is the most important aspect of hosting such a data intensive application.