**Score and Threshold Manipulation**  
  
The prior step has made sure to extract all possible relevant GIFs from the dataset. The NLP and ML techniques does a very good job to find even the most remotely relevant GIFs to the input message, but it fails to distinguish between a good and a bad GIF. In other words, it has no idea which GIF is more popular, more relevant, or more likely to get chosen by the user. The model also has no filter to limit the number of outputs. Say an input, “Hi! What’s up?” will have 100s of relevant GIFs in the dataset, while an input such as, “Are you going for meeting tomorrow?” may have less matches. We need to make sure we display a reasonable amount of filtered out options to the user and not flood them with GIFs.  
  
On the other hand, no matter how strong the NLP algorithm is, it fails to develop a learning curve. It fails to learn from user’s prior decision and interests. To make an adaptive model which learns from the user’s behavior needs more than just tokenization of messages.   
  
Now that we have a rough idea on what our final step should perform before showing options to the user, we can come up with different hyper-parameters for the same.

* We can create a threshold which can act like a filter to select GIFs with a better score.
* We can periodically change GIF’s scores by a certain amount, according to user’s behavior, a higher score reflects that the user likes a particular GIF.
* We can set an upper limit for the number of GIFs displayed to the user.

The above discussed hyper-parameters promise to solve many of our problems discussed above, but they come with new challenges.

* What is our initial threshold and score for a new user?
* Should we keep a high threshold for better accuracy or a low threshold to ensure regular variety in the options?
* Do we need to change the threshold over time looking at user’s usage?
* Does the score change equally for all scenarios?
* Is threshold/change in threshold same for a new and an old user?
* What if new GIFs are added over time?
* Is no output better than irrelevant output?
* How to ensure no GIF is ignored in spite being disliked by the user?

To achieve an optimal model, we need to make sure that all the above questions are answered optimally. The hyper parameters need to be selected carefully such that the learning curve is not too steep nor too flat, in other words, the best possible trade-off needs to be made.   
  
Keeping all the problems and loop holes in mind, we wish to create an algorithm that satisfies all our questions and does justice to the NLP algorithm model created. We will be implementing our solution in python and attaching to the prior step to form a smooth backend for the entire program. The goal is to create a smart model for the user and to ensure that the user has no or minimal tasks after giving the input.