**Comments for kmeans\_cluster\_email.py**

You have attempted to use k-means clustering and PCA. As you are aware k-means clustering is an unsupervised algorithm, when you get two clusters you are not sure that the two clusters are hams or spams. It is good idea to consider various classifications algorithms that we had picked up like SVM and ensemble models. Please try them and provide re**vised code**

**Comments dated 23rd May 2020**

**Comments for Email Classification Using LSTM**

You have used bidirectional lstm to solve ham vs spam this is a good idea as it gives you good results . It is not clear why you have to do tfidf vectorizer before passing the inputs to the embedding layer. Neural networks do not need lot of feature engineering and the embedding layer will convert the words to meaningful vectors. Try eliminating tfidf vectorizer and see if you are getting any drop-in performance

It would be a good idea to try out pre trained models like BERT or flair on this data set. This will reduce the overall cost of training as we don’t need to start all the way from scratch

Once you test this you want to consider using tensorflow serving, kubeflow or AWS sagemaker for deployment and expose the prediction A**PI**

**Comments for Catsdogs.py**

I see that you have used multiple conv2d and maxpooling like we see in our example how much accuracy decrease are you facing if you reduce number of layers. You can use packages like hyperopt for identifying hyper parameter tunning. You can additionally use vgg net when you trying to identify cat or dog faces. Please try making changes using pre trained models example for those also present in our repository.

After you test all of this you can setup a pipeline using tensorflow serving or AWS sagemaker or kubeflow.