Milestone 4 Report

Final state

Our project is able to run locally and identify users' mood accurately and accordingly display Spotify playlists that we have created. We have found and fixed minor bugs and completed our web frontend design. Our web backend is able to receive a request and image data from the frontend. The backend then manipulates the image and passes it through different machine learning components to predict users' current mood. We were faced with an obstacle about deployment that was not able to be solved in a timely manner (further discussed below).

Several methods were tested to fix the overfitting problem of our training model: adding dropout layers, adding more transforms into the data loader (random crop, random rotate, horizontal flip). However, we were not able to fully solve the overfitting problem, yet we achieved a good validation accuracy of 86-88%.

Final Obstacles

The main challenge that we faced was deploying the website using Google Cloud. The web app is able to run locally, but not on Google Cloud. According to the app engine guide, "If a script handler generates a response larger than this limit, the server sends back an empty response with a 500 Internal Server Error status code." The build logs stated: "While handling this request, the process that handled this request was found to be using too much memory and was terminated. More information about the obstacle is stated in Steve's section below.

Team member's tasks

1. Deepan Chakravarthy

As per the previous goal set, I worked on regularization of the model to fix the overfitting in the training. Several methods were tried. First, one dropout layer was added before the last linear layer. The validation accuracy dropped, however the testing accuracy remained at around 99% even after 20 epochs. Next, all the gaps in between the linear layers in the sequential container were filled with dropout layers with parameter 0.5. So, the fc layer consisted of: 'dropout-linear-relu-dropout-linear-relu-droput-linear'. However, this further reduced the accuracy to sub 80% while the test accuracy remained unchanged.

Online sources claimed that overfitting tends to occur if the training dataset is smaller than the ideal size for the problem at hand. To solve this issue, I added more transforms into the data loader, so that over multiple epochs there would be more different samples. Specifically, random crop, random rotate, horizontal flip. However, this only led to the training accuracy reducing by 3-4%. Overall, the issue of overfitting is not solved, yet a good final validation accuracy of 86-88% is reached.

2. Aryan Gandhi

For this milestone I worked on some of the front end web development, code cleanup, and with Steve attempted to deploy the django web application using Google cloud. For the front end web development, I wrote the about section of the website, listing why we decided to work on this specific project and how the application works. In addition to that I worked on cleaning up the code, removing unnecessary lines and deleting files that were not being used. In an attempt to deploy the application we were able to allow it to build but were received with a 500 Internal Error status code on the website link. We decided to put more focus on other parts of the project and ensure the website was working correctly locally rather than focussing on deploying it.

3. Steve He
I made some minor changes to the website and tried to deploy our django website on google cloud.

I looked into the log in google cloud from Aryan’s deployment and found the problem and after Aryan resolved it, it was built successfully. However we met a server error and I found out that is because google cloud app engine have a dynamic responses limited. According to the app engine guide: ”If a script handler generates a response larger than this limit, the server sends back an empty response with a 500 Internal Server Error status code.”

And I also find this in our log: “While handling this request, the process that handled this request was found to be using too much memory and was terminated. This is likely to cause a new process to be used for the next request to your application. If you see this message frequently, you may have a memory leak in your application or may be using an instance with insufficient memory. Consider setting a larger instance class in app.yaml.” Followed by: “Exceeded soft memory limit of 256 MB with 428 MB after servicing 0 requests total. Consider setting a larger instance class in app.yaml. ”

So I manually set our instance class to F2 to fit the usage of memory and it successfully builds but it runs with an AttributeError: module 'keras.utils.generic_utils' has no attribute 'populate_dict_with_module_objects'. This error did not appear on our local machines. And I also tried to change the version of keras and tensorflow but this error still came up.

4. Arya Phan
For this milestone, I completed the final designs of our web pages, cleaned up code, and prepared slides for our video, and a diagram for our final report. I improved the user interface for our 3 web pages by changing the fonts and the navigation bar appearance and adding all the necessary links to make sure the 3 web pages are connected. For the About page, I finalized the design and summarized what our project is about and what software tools we used to implement our web app. For the output webpage, I generated messages to users based on the predicted result from the model. Additionally, we have discussed attempts to fix our deployment issues described above but we decided to focus more on other components of the project and make sure everything works properly locally.