Project Recap from D1:

From Open Access Series of Imaging Studies (OASIS) dataset, we have chosen which is a set of MRI scans for Alzheimer disease patients as longitudinal studies. We would use “Alzheimer preprocessed dataset” by Sachin Kumar on Kaggle which is a cross-sectional dataset and MS: “Multiple sclerosis” by Burak TAŞCI and Brain MRI Dataset of Multiple Sclerosis with Consensus Manual Lesion Segmentation and Patient Meta Information data for preprocessing.

Data Preprocessing: The dataset we chose to work with are MRI scans of the brain. The initial set looked more taken from a single source, specifically the OASIS dataset and Multiple Sclerosis dataset by Burak Tasci but we enlarged the whole dataset by scouring medical data from various sources on MRI imaging of brain. There are four categories we are going with: very mild, mild, moderate and no dementia data each with over 1000 MRI scans.

ML Model: Our model loads all the images from the folder and uses the training model implemented to learn and train more on the classification of the different types of MRI scans. To achieve this, we use Py Torch, OS, NumPy and Torchvision libraries.
We focused on using Convolutional Neural Networks where it learns spatial hierarchies and capturing local patterns in the data. From CNN, we used ResNet to address and gradient problem. This has been highly effective for image processing tasks. There are 32 layers that we are currently working with.
We decided to work with a large dataset, to improve generalization. To optimize the dataset, we implemented loss functions so that the max likelihood of loss does not affect the overall quality of data. We had to keep in mind of the batch data, mostly not making it so big that it can affect memory and create issues on generalization.

There were a lot of memory and optimization issues since we had to deal with a very large dataset. Although, we managed to reduce a bit using libraries and vectorizing the operations, we are still working on the best way to handle all the data without majorly impacting on memory and training data.

Preliminary Results: The model can successfully generate the training data sets using the CNN we created. As of now, the accuracy of the validation and test set is at 50%, although we are not sure why is it so low. We have also fine-tuned and optimized our hyperparameter to our model to classify the datasets on different labels. However, we are into some time-complexity issues where the program runs for too long. In near future, we will think about what we could further do to resolve this problem.

Next Steps:

Our user will submit an image from an MRI. Our ML algorithm will analyze whether the brain has one (or more) of the diseases that the algorithm was trained on. It will continue to identify whether the individual is at-risk for any other diseases. We will also highlight the anomalies in the MRI. We need to make better assumptions and generalizations on what and how much data the user wants to be processed. So, how can we use the dataset to better optimize the model to handle any new data, whilst making the model as efficient and time effective as possible is a challenge.