CS6240 Project
Name: Rushabh Shah
CCIS Github Repo: https://github.ccs.neu.edu/rushabh0812/Project

1. **Type of model used and parameters explored for it.**

Ans: We have used Random forest classification model of Spark-MLLib library to classify an image pixel as foreground and background.

Since Random Forest Classifier model is an ensemble of decision trees, we varied the number of trees, the maximum depth of the tree of our model.

2. **Pseudo-Code**

- Read images into memory (RDD)
- Convert image values from String to Double
- Sampling the data using some as training data and some images as validation data
- Using Random Forest algorithm on training data using parameter values of tree depth and maximum trees.
  val model = RandomForest:trainClassifier(trainingData; new Strategy(algorithm; impurity;maximumDepth); treeCount; featureSubsetStrategy; seed)
- Testing the model on validation data
- Run the model on testing data and report the predicted output

2a. How many tasks are created during each stage of the model training process?

Training: Equal to the number of decision trees times depth of tree, since Random forests train a set of decision trees separately, so the training can be done in parallel.

Prediction: Coalesce one spits only a single file which means single task otherwise 97 tasks are executed.

2b. Is data being shuffled?

Training: Yes, Random Forrest injects randomness into the training process hence the data is shuffled.

Prediction: No, as we do the prediction on a trained model.

2c. How many iterations are executed during model training (for methods that have multiple iterations)?

Training: One
Prediction: One

2d. How did changes of parameters controlling partitioning affect the running time?

Training: Max depth and tree count are the parameters that control the running time. Below there is a table which shows that.

Prediction: It does not affect the prediction phase as the model has already been trained.

3. **Pre-processing:**

Major pre-processing steps that were followed are:

- Converted the format of Image values from string to double to be used in Random Forest algorithm.
- Since this an image recognition problem, and every row of data describes an unique pixel of the brain scan image, we took all the features to train our model.
- Created a LabeledPoint containing entire neighborhood vector as Feature and foreground/background binary value as Label
- The inclusion of Gini Impurity parameter of Random Forest model takes care of the un-homogeneity of the labels (The frequency of 0s and 1s) while training the data.

4. **Accuracy numbers for different parameter settings you explored.**

Max Depth	**Tree Count**	**Accuracy**	**Time**
3	**150**	**0.9965**	**1806**
3	**200**	**0.9966**	**1688**
5	**150**	**0.9944**	**2011**
5	**200**	**0.9946**	**2032**
6	**150**	**0.9940**	**2070**
6	**200**	**0.9936**	**2870**

5. **Running time for the model training and the prediction phase**

Tree Depth = 6
Tree Count = 200

Machines	**Model Training**	**Prediction**
10	**2211**	**504**
12	**2019**	**438**