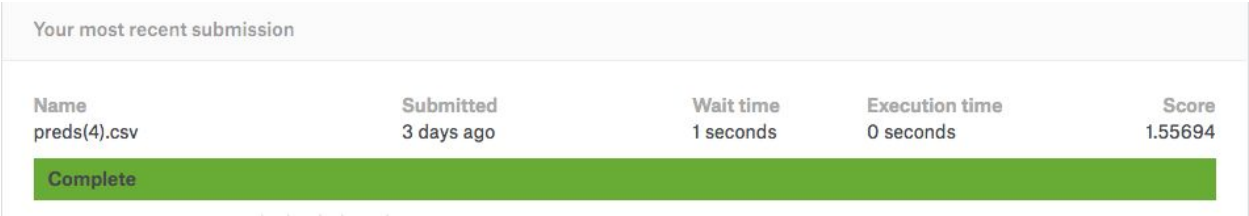
Colab



Problem Definition: This weeks objective was focused on implementing a Convolutional Neural Network (CNN) to accurately distinguish between images of dogs and cats. The dataset was comprised of 25000 training images and 12500 test images of varying sizes. Out of sample predictions were submitted to Kaggle for evaluation; results were scored using a log loss metric. Insight gained from this experiment will be used to establish a broad set of guidelines and recommendations to maximize image classification accuracy.

Research Design/Programming: A compressed ZIP file containing the 25000 training instances was uploaded to Amazon S3 hosting. The ZIP file was loaded and each of the 25000 images were decompressed and resized to 200 x 200 pixels. The resized images were then converted into numpy arrays (200,200,3). The appropriate labels were assigned to each image by extracting "dog" or "cat" from the file name. Labels were then converted to 0 for cat and 1 for dog.

Several Convolutional Neural Networks were implemented using 10 epochs. Each model was evaluated using training accuracy, validation accuracy and out of sample log loss score. Computational resource limitations prevented the use of RandomSearchCV or GridSearchCV for hyperparameter tuning. A workaround was used to combat this issue. Instead of running

models in succession, a model was run and then serialized using pickle so that it could be loaded at a later point to make final predictions on out of sample data. The model returns the probability that an image is of class 1 = Dog.

	Train_acc	Val_acc	log_loss_score
model1	0.879289	0.8648	1.81967
model2	0.879733	0.7428	1.86263

Recommendations for Management:

Results were inconclusive. What we did learn from this exploratory effort was that our computational resources are not currently sufficient for deep learning. More GPU and RAM will be needed moving forward. Amazon Web Service (AWS) and Google Cloud Platform (GCP) both offer affordable access to Deep Learning resources. Similarly, moving forward we could implement python generators instead of standard iterables to decrease memory usage.

One way to improve model accuracy would be to obtain more training data. However, this can be a time consuming and costly investment. To mitigate the upfront cost of gathering more data we could distort and manipulate each training image, ultimately increasing the size of the training set.