CS6240
Final Presentation

Team member:
Chandrika Sharma
Meng Tao

Random Forest

The Random Forest (RF) algorithm is a suitable data mining algorithm for big data. It is an ensemble learning algorithm using feature sub-space to construct the model. Moreover, all decision trees can be trained concurrently, hence it is also suitable for parallelization.

Solution

We trained 8 separate models with Random Forest Classification

Performed Bagging to get majority-vote label

SPEED UP(per model)	Training Time (min)	Prediction Time (min)
6 m4.large	54.6	4.7
11 m4.large	22.4	4.6

EXPERIMENTATION

Sparse Vs Dense

Since data had more 0s than 1s, we created Sparse Vectors as feature input to the RF Model

Preprocessing to create these Vectors took a lot of time and ran for hours.

Converted the data into dense vectors and the process sped significantly

Why?

Preprocessing time too high to convert to Sparse Vector

Data Diversity

With tuning the model we observed that the accuracy didn't improve too much and that's when we decided to add more diverse data as suggested in the optional part of the project requirement. But….

All the data into one model caused

1. OOM errors
2. Hours of execution
3. Hence, costly in time and money

Solution??

Bagging !!!

Persist Data? Where?

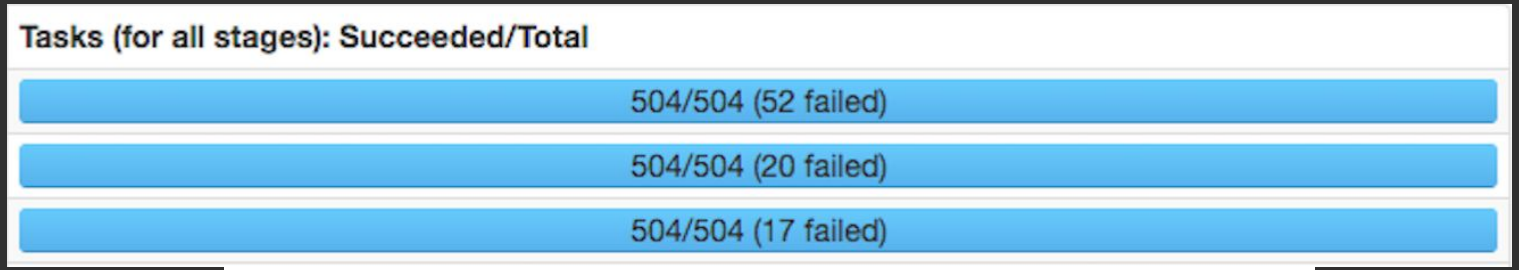
Creating the additional data points involved multiple operations on the few RDDs

Persist?

We didn’t want to persist in-memory(didn’t want to compromise on computation speed) and decided to go with persisting in DISK ONLY.

DISK ONLY Persist

On switching to DISK, we ran 3 models on 15 machines with persisting, and saw tasks fail continuously.



Executors

▸ Show Additional Metrics

Summary

	RDD Blocks	Storage Memory	Disk Used	Cores	Active Tasks	Failed Tasks	Complete Tasks	Total Tasks	Task Time (GC Time)	Input	Shuffle Read	Shuffle Write	Blacklisted
Active(15)	417	46.6 GB / 42.6 GB	45.4 GB	56	5	0	6562	6567	9.0 h (2.0 h)	568.5 GB	3.7 GB	5.2 GB	0
Dead(1)	31	3.3 GB / 2.8 GB	3.6 GB	4	0	0	521	521	40 min (7.8 min)	42.2 GB	360 MB	346.5 MB	0
Total(16)	448	49.9 GB / 45.4 GB	48.9 GB	60	5	0	7083	7088	9.6 h (2.1 h)	610.8 GB	4 GB	5.6 GB	0

What to do?

Then we decided to unpersist the data as soon as it’s done with computation and saw successful execution.

Optimization/Future Works?

Confusion Matrix: imbalanced

Total: 978344	Predicted: 0	Predicted: 1
Actual : 0	971553	495
Actual : 1	2528	3768

Solution : Over sampling or Under sampling! We could use the rotated images to increase data with the foreground points (with label 1).