Defect Identification

Aruna Gunda
Nathan Kot
Sreesha Nath
Tian Zhang

Why is the project worth studying?

Hard drives are an integral part of the storage ecosystem and all of us interact with one on a daily basis. Industry analysts have forecast a memory shortage to occur somewhere in the 2020s. Advances in IOT, picture/video standards, mobile storage, etc., have put considerable strain on memory companies; not a bad thing when memory is your business. Cloud providers cannot afford to double their footprint as demand doubles, else they would go out of business. Instead they rely on themselves and suppliers to come up with novel techniques to provide margin. One of these “novel” techniques is to shove more disks into the same form factor.

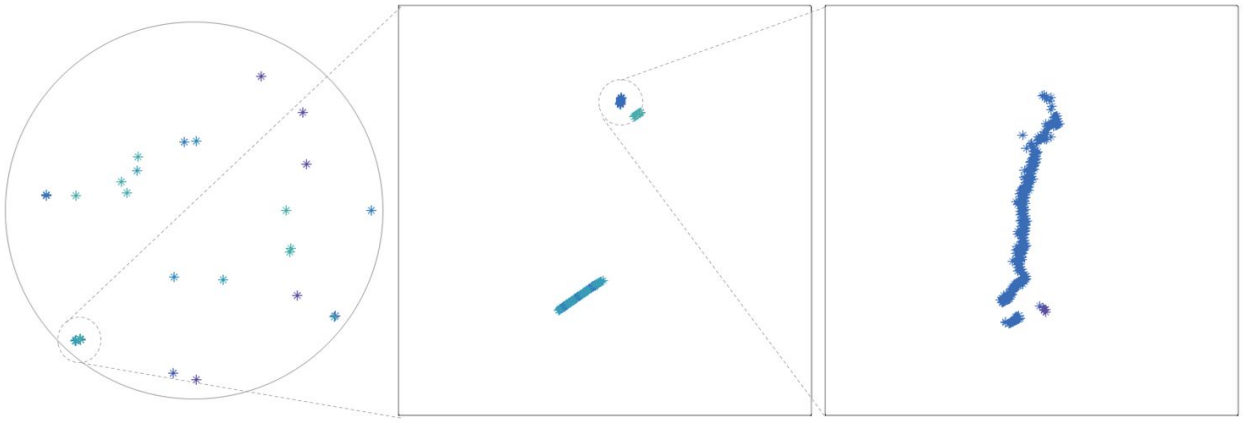
Each hard drive is a set of individual disks. It so happens that during the manufacturing process, due to machine or human error, there are segments that get damaged. These defects range from surface roughness to contamination. Such defects render those areas unusable. Too many of these defects on a surface and the disks are no longer considered viable.

The most confident detection and identification of these flaws and their severity is done at the last stage, at a time when the hard disk is completely assembled. This often results in the HDD being unshippable, and thus a large expense. The primary motivation of our project is to detect such flaws at an earlier stage so that unviable disks can be filtered out prior to their being built into a drive. Especially with helium-filled drives becoming standardized as every drive is now welded shut. This would make drives of better quality and cost the company less, as there would be fewer drives that would be need to be discarded as waste.

Why machine learning techniques are appropriate

Our problem is about finding whether or not a disk is viable enough to go to the market. The data forming our feature set is that obtained by various sensors prior to the disk being built into drive. The data forming our label set is that obtained after the drive has been assembled, when confidence in defect identification is at its highest and most concise. With the poor quality upstream information we have, it is not possible for humans or any non-computational models to identify whether a feature set point cluster is a true defect or not. As the relation between the sensor data and the outcome is mathematical we need computationally intensive models that are clever enough to identify the patterns in the data, group the data points into clusters and identify whether or not they are true defects. Thus at every stage of our project, we need machine learning techniques like clustering to identify groups and efficient supervised learning methods like neural networks to identify complicated patterns and classify clusters as defective or non-defective.

Demonstrate that you (will) have access to datasets



Currently Seagate can and does produce millions of disks per month. Each of these disks produces hundreds of sets of sensor data, MBs each, throughout their lifecycle. There is more data available than is realistically tenable. Sifting through a week's worth of local (design center, not production) data yielded 2GBs.

For this project we have chosen a subset of this sensor data to use as our feature sets. The sets will be constrained to a single, mature product such that configurations affecting the character of said data will be fixed across samples.The labels of said feature sets are produced after the drive has been assembled and calibrated, during final quality assurance testing. Here time is sacrificed for confidence but the results mean life or death for the drive.

Initial thoughts of possible approaches

The requirements for our project can effectively be split into three separate tasks: identification of groups of data points likely to be associated with the same defect, alignment of these groups with in-drive data (labels), and finally prediction of whether any particular group of points is indicative of a true defect or just insignificant noise. To accomplish these tasks, our initial thoughts are to use a clustering algorithm (likely a modification of DBSCAN or similar) to identify groups of points, ideally with stronger sensor hits near the middle and weaker hits near the edges. Once we have a list of clusters, we plan to roughly align them with some kind of distance based algorithm. Ideally, this would be as simple as matching each cluster to its nearest neighbor. If this is insufficient, we plan to implement an automatically generated alignment where the cost function is some combination of the total distance between clusters to in-drive points (probably without any kind of power and some sort of upper penalty limit, since a fair amount false positives are to be expected), and with some penalty based on the amount of rotation/translation needed for the alignment (since we are expecting their alignments to be fairly close to begin with). Finally, we'll have a lists of sensor hits and associated labels (or lack thereof), which can be fed into a supervised learning method, such as neural nets or logistic regression, the specifics of which will depend on how the data density of the clusters look after grouping.