# Hard Drive Failure Prediction

Project write-up for the course Data Science for Business Analytics at Stern School of Business, New York University

By

Tushar Bhatkal
Satyajeet Maharana
Varsha Murali

# ABSTRACT

Data centers can largely benefit from a service that employs data mining to predict hard drive failures. Although hard drive failures are rare, they are costly occurrences. Failures in hard drives could result in temporary system unavailability and/or data loss. Hard drive manufacturers use Self-Monitoring and Reporting Technology (SMART) attributes collected during normal operations to predict failures. These SMART attributes report daily diagnostics of hard drives such as read/write error rates, spin retry count, power cycle count, etc. We used publicly available data from Backblaze, who started recording the stats of a large number of hard drives (~47000) from their own data center. In this project, we analyze and compare the performance of various machine learning algorithms (Linear Regression, Decision Tree, AdaBoost, XGBoost, Gradient Boosting and k-Nearest Neighbors) when used to predict hard drive failures using Backblaze data in the year 2018. [1]

# I. INTRODUCTION

In this burgeoning digital age, data is being created at an astounding rate. With storage media capacity increasing and data storage costs plummeting, businesses are now collecting data, both trivial and critical, in hopes that it may someday be of value. As a result, data centers are expanding, and hard disk drives are their dominant storage technology. It is an increasing challenge to maintain storage system reliability since manual monitoring is far from practical considering the size of the average data center today.

It appears to be useful if a model can be developed to predict hard drive failures and then the data center operators might use the prediction result to improve system reliability. There are some challenges which have to addressed in using this approach to solve this business problem. First, hard drives are highly reliable devices, implying that their failures are rare events. This would mean that the dataset used to train models is going to be skewed with a high ratio of healthy to unhealthy hard drives. This is likely to affect a model's performance as there may not be much of a pattern that can distinguish the two classes of hard drives. Second, since there are a few thousand hard drives in a data center, the dataset is quite large, resulting in high computational cost in model training. Third, it is important to keep in mind that it is imperative that the false positive rate is low upon using a model. [2] Here, a false positive is a healthy hard drive being classified as unhealthy.

The subsequent chapters discuss the project in detail. Chapters 2 and 3 involve details about the dataset used and how the predictor variables were chosen to be used in model training. The various models used to solve the problem and their respective performance metrics are discussed in Chapter 4. In Chapter 5, an overview of the generalized model performance and the return on investment for the data center is provided. Chapter 6 talks about how our proposed solution can be deployed in the real world. Concluding remarks are made in Chapter 7.

## II. DATA UNDERSTANDING

In this project, we used hard drive data made publicly available by Backblaze, an online backup and cloud storage provider. Backblaze provides a daily snapshot of each operational hard drive in their data center from 2013 to Q1 of 2019. The daily snapshot reports a hard drive's health diagnostics as 90 SMART attributes, both raw and normalized values. For the purpose of this project, we have resorted to using the data from 2018. As a result, the dataset we chose to work with had approximately 33 million rows and 105 columns.

The normalized SMART values are vendor specific and since the Backblaze data center has hard drives from 4 different manufacturers (Toshiba, Seagate, Western Digital and HGST), we only consider the raw data columns [1]. Furthermore, each manufacturer supplies different hard drive models to the data center. In order to have consistency in the attributes reported, we decided it be best to work with one particular model by a manufacturer. This also greatly reduced the size of our dataset. How we came to choose the model and manufacturer is discussed in Chapter 3. After an extensive literature survey, we were able to deduce that not SMART attributes were predictive of failure. [3] The attributes that were initially considered to work with are listed in Table 1.

*Table 1. Column names and their respective attribute names for predictive SMART stats*

| Column Name | Attribute Name |
|---|---|
| smart_raw_5 | Reallocated Sectors Count |
| smart_raw_10 | Spin Retry Count |
| smart_raw_12 | Power Cycle Count |
| smart_raw_184 | End-to-End error / IOEDC |
| smart_raw_187 | Reported Uncorrectable Errors |
| smart_raw_188 | Command Timeout |
| smart_raw_189 | High Fly Writes |

| | |
|---|---|
| smart_raw_190 | Temperature Difference or Airflow Temperature |
| smart_raw_196 | Reallocation Event Count |
| smart_raw_197 | Current Pending Sector Count |
| smart_raw_198 | (Offline) Uncorrectable Sector Count |
| smart_raw_199 | UltraDMA CRC Error Count |
| smart_raw_200 | Multi-Zone Error Rate |
| smart_raw_201 | Soft Read Error Rate or TA Counter Detected |

After picking the above 14 features to work with based on our readings, we built models to predict failures. However, the results weren't too impressive. So, we decided to further analyze the features to check which of them, if any, showed large variance in unhealthy hard drives when compared to healthy hard drives. Out of 24,432 hard drives in the dataset, 121 hard drives failed in 2018.

It was observed that most of them followed the trend for SMART values 12, 196, 198 and 201 as shown in Fig.1 (a) and (b).

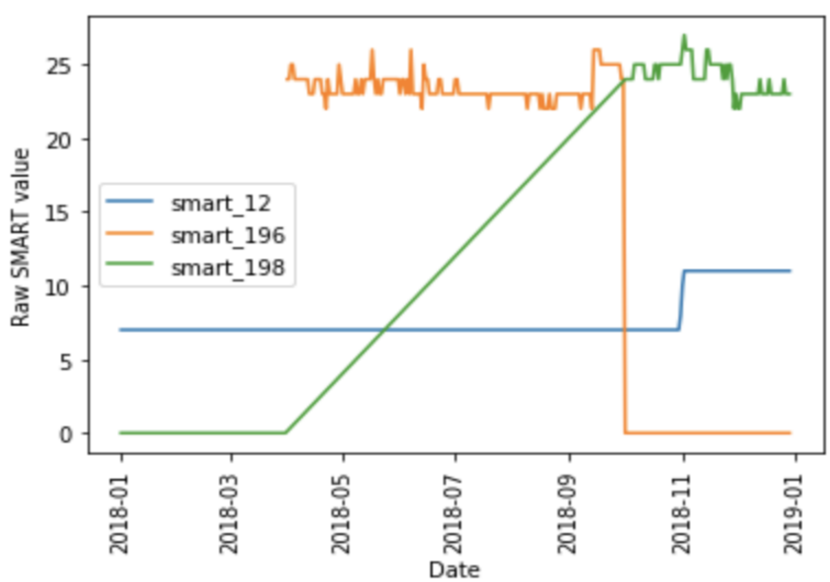


*Fig.1 (a) Variations in SMART values 12, 196 and 198 in failing hard drives*

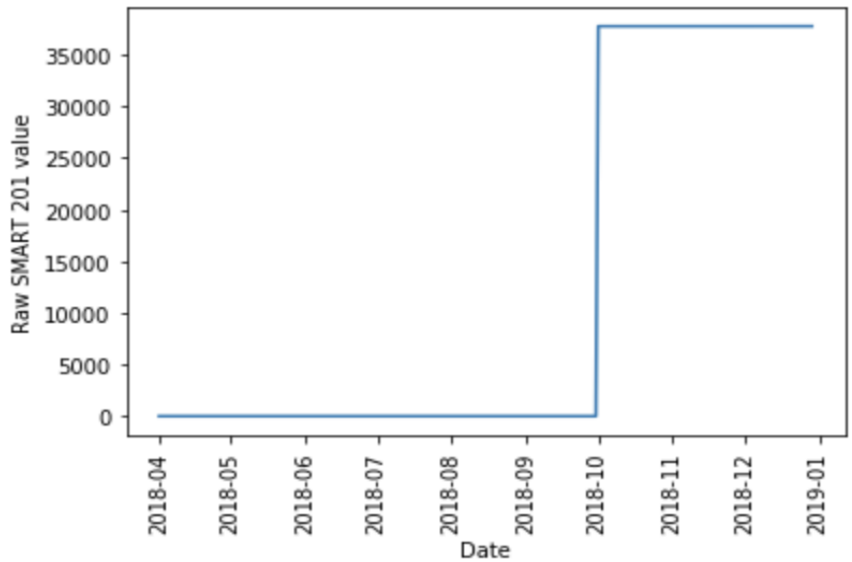


*Fig.1 (b) Variation in SMART value 201 in failing hard drives*

From Fig. 2 (a) and (b), it can be seen that the SMART values 12, 196, 198 and 201 barely vary or are null during some parts of the year for healthy hard drives.

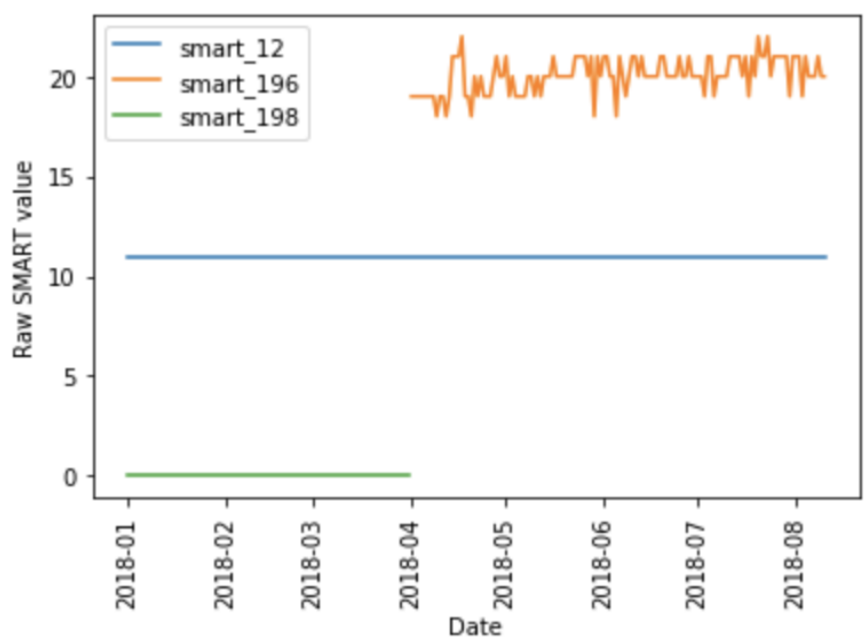


*Fig.2 (a) Trends of SMART values 12, 196 and 198 in healthy hard drives*

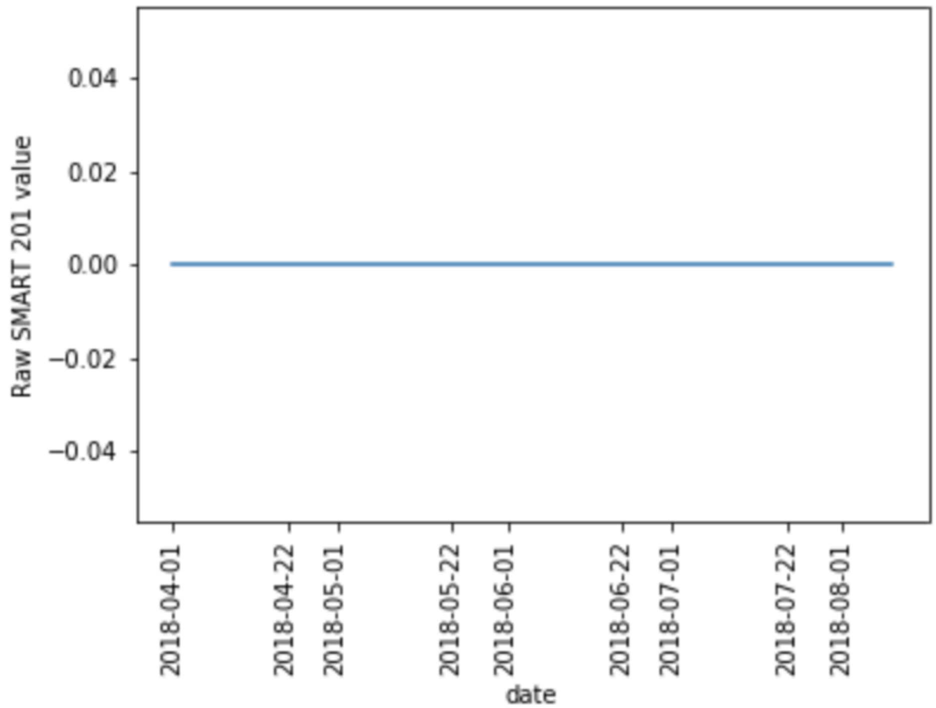


*Fig.2 (b) Variation in SMART value 201 in failing hard drives*

After performing the above preliminary analysis to understand our dataset, we chose SMART values 12, 196, 198 and 201 to build our models on.

# III. DATA PREPROCESSING

The data obtained from Backblaze is made available publicly as csv files for each day. We used data from 2018 for our analysis. The csv files for each of the 365 days were read and concatenated into a dataframe using Scala.

In order to identify which manufacturer and model to pick for the purpose of our study, we analyzed the data from Q4 of 2018. Table 2 shows the number of hard drives per manufacturer and the number of failing instances.

*Table 2. 2018 data broken down by manufacturer*

| Company | Total Hard Drive Count | Failing Hard Drives | Percentage of Failure |
|---|---|---|---|
| HGST | 20814 | 26 | 0.1249 |
| Seagate | 83849 | 341 | 0.4066 |
| Toshiba | 2288 | 21 | 0.9178 |
| Western Digital | 683 | 5 | 0.7320 |

Although Seagate has a lower percentage of failing instances compared to Toshiba and Western Digital, it was chosen because more data is available on Seagate drives that fail. Amongst the Seagate drives, it was found that model 'ST4000DM000' was the most common. As a result, we eliminated hard drive instances of all other models and manufacturers other than Seagate 'ST4000DM000'.

Next, for each failing hard drive, the 'failure' column for *n* days preceding the actual failure date is changed to 1. Here, *n* is chosen to be 45. It is also observed that for SMART values 196 and 198, there are null values. Certain instances belong to hard drives which upon failure become null. In other cases, these smart stats (which are error counts) remain null until errors are encountered in the hard drive. Thus, for the purpose of building our model, instances with null values are discarded.

# IV. DATA MODELING AND MODEL ANALYSIS

**Performance Metrics**

In this Chapter, we specify the models built and the results obtained from using each of these models on the test data. Before that, we list the various performance metrics that were used to analyze the performance of each model.

Below is the split of healthy and unhealthy hard drives in the dataset for the year 2018:

*Percentage of Healthy Hard disks: 99.50%*

*Percentage of Unhealthy Hard disks: 0.49%*

*Total Number of hard disks: 24432*

Since our data is highly skewed, pure ‘vanilla’ accuracy is not going to be a valid estimator of a model’s performance. This is because if a model simply predicts ‘healthy’ for all hard drives in the dataset, the accuracy is going to be extremely high (99.50% since that is the percentage of healthy hard drives). Therefore, we rely on other evaluation metrics to compare the different models. Listed below are the metrics used:

1. Recall or Sensitivity or True Positive Rate:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

2. Precision or Positive Predictive Value:

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

3. Fall-out or False Positive Rate:

$$False\ Positive\ Rate = \frac{False\ Positives}{False\ Positives + True\ Negatives}$$

4. Miss Rate or False Negative Rate:

$$Miss\ Rate\ = \frac{False\ Negatives}{False\ Negatives\ +\ True\ Positives}$$

5. F1 Score:

$$F1\ Score\ = 2\ \cdot \frac{Precision\ \times\ Recall}{Precision\ +\ Recall}$$

6. Receiver Operating Characteristic (ROC) Curve: Since, the operating conditions like the costs and benefits are unknown to us at this point, we would need to evaluate our model performance using the ROC graphs. In a ROC curve, the true positive rate is plotted as function of the false positive rate for different cut-off points. Each point on the ROC curve represents a sensitivity/specificity pair corresponding to a particular decision threshold. A test with perfect discrimination (no overlap in the two distributions) has a ROC curve that passes through the upper left corner (100% sensitivity, 100% specificity).

7. Area under the ROC Curve: In order to be able to compare different models, we would also get a numerical value of the ROC curve, which is known as the area under the ROC curve (AUC).

**Performance Analysis**

The dataset was split into two sets, training and testing, using sklearn's train_test_split function with the 'stratify' parameter set to the target variable. This was done to maintain the balance of healthy to unhealthy hard drives the same in the training and testing sets for the purpose of building a good model and analyze their performance on an equally balanced holdout set.

We realize that just one split of the data into training and testing may not provide generalized results. We later perform cross validation tests to get some general statistics on model performances. Following are all the models implemented for the dataset and their respective results.

### 1. Logistic Regression

| | |
|---|---|
| Accuracy on Test Data: | 0.217 |
| Recall: | 1 |
| Precision: | 0.00218 |
| False Positive Rate: | 1 |
| F1 Score: | 0.0043 |

| Confusion Matrix | | Predicted 0 | Predicted 1 |
|---|---|---|---|
| Actual | 0 | 0 | 753199 |
| | 1 | 0 | 1645 |

ROC Curve:

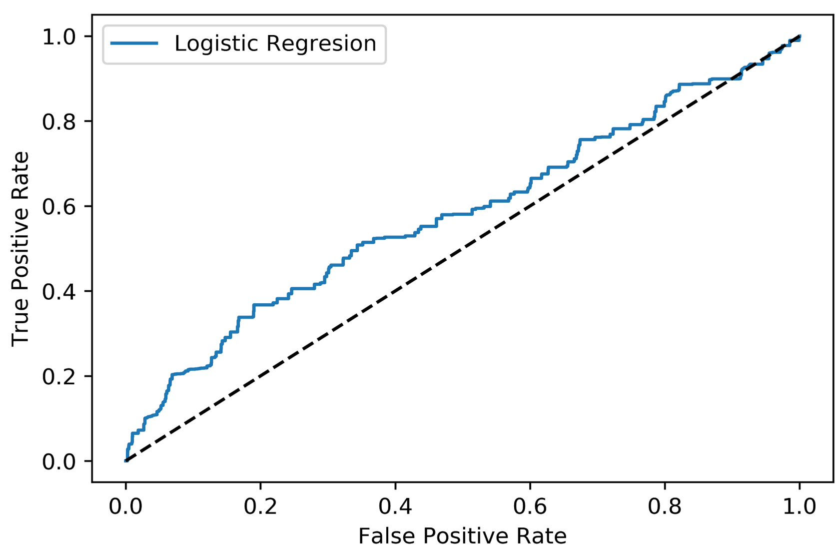


*Fig.4 (a) ROC Curve for Logistic Regression with AUC = 0.581*

Our confusion matrix reports 'instances' of hard drives that have been correctly classified as attributes of a healthy or failing hard drives. It can be seen that simple Logistic

Regression performs very poorly. It classifies all hard drive instances as belonging to an unhealthy hard drive. The model has failed to mine any useful pattern to distinguish the two classes.

### 2. Decision Tree Classifier

| | |
|---|---|
| Accuracy on Test Data: | 0.99 |
| Recall: | 0.897 |
| Precision: | 0.89 |
| False Positive Rate: | 0.0002 |
| F1 Score: | 0.893 |

| Confusion Matrix | | Predicted 0 | Predicted 1 |
|---|---|---|---|
| Actual | 0 | 753017 | 182 |
| | 1 | 170 | 1475 |

ROC Curve:

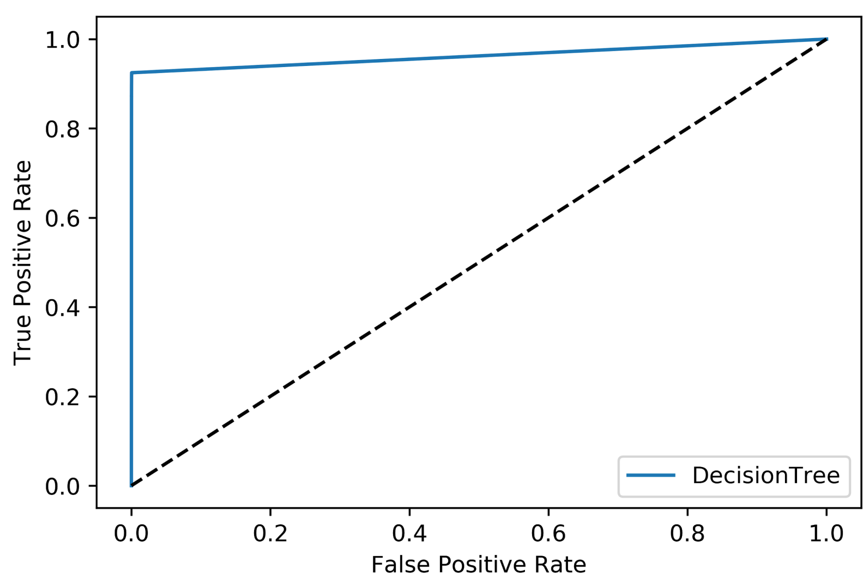


*Fig.4 (b) ROC Curve for Decision Tree Classifier with AUC = 0.962*

The Decision Tree classifier performs exceedingly well with most instances classified correctly. Although there are some instances classified falsely, the classifier identifies more truly unhealthy hard drives as compared to Logistic Regression.

### 3. Adaboost Decision Tree Classifier:

| | |
|---|---|
| Accuracy on Test Data: | 0.99 |
| Recall: | 0.899 |
| Precision: | 0.9 |
| False Positive Rate: | 0.0002 |
| F1 Score: | 0.9 |

| Confusion Matrix | | Predicted 0 | Predicted 1 |
|---|---|---|---|
| Actual | 0 | 753036 | 163 |
| Actual | 1 | 165 | 1480 |

ROC Curve:

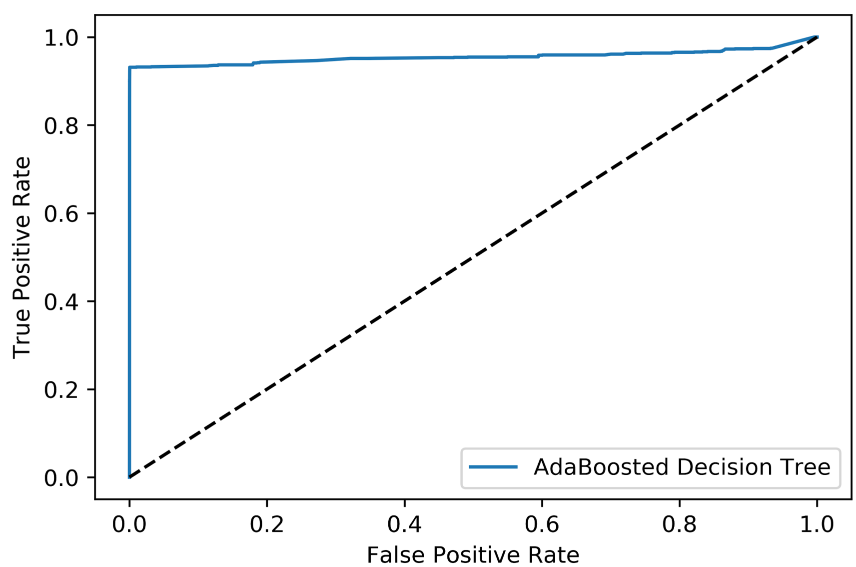


*Fig.4 (c) ROC Curve for AdaBoosted Decision Tree Classifier with AUC = 0.955*

The next model we tried was an AdaBoosted Decision Tree Classifier. This model performs better than the simple Decision Tree Classifier. The reported false positive and false negative values are much lower than that of the Decision Tree built previously.

**4. Gradient Boost Classifier:**

| | |
|---|---|
| Accuracy on Test Data: | 0.99 |
| Recall: | 0.342 |
| Precision: | 0.767 |
| False Positive Rate: | 0.0002 |
| F1 Score: | 0.473 |

| Confusion Matrix | | Predicted 0 | Predicted 1 |
|---|---|---|---|
| Actual | 0 | 753028 | 171 |
| | 1 | 10835 | 62 |

ROC Curve:

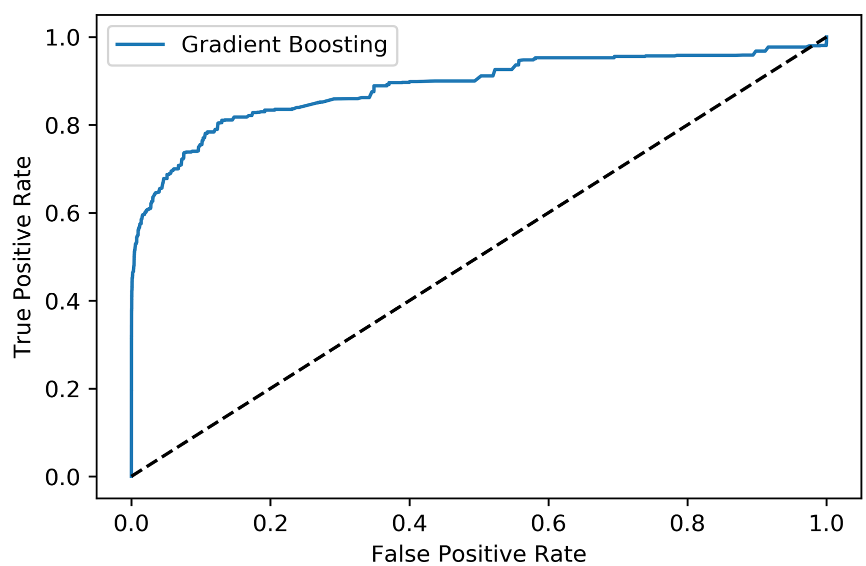


*Fig.4 (d) ROC Curve for Gradient Boosting Classifier with AUC = 0.887*

Gradient Boosting results are not as good as the Decision Tree or the Boosted Decision Tree. It has some trouble identifying unhealthy hard drives since the false negative cell in the confusion matrix is high.

### 5. XgBoost Classifier:

| | |
|---|---|
| Accuracy on Test Data: | 0.85 |
| Recall: | 0.613 |
| Precision: | 0.009 |
| False Positive Rate: | 0.149 |
| F1 Score: | 0.0175 |

| Confusion Matrix | | Predicted 0 | Predicted 1 |
|---|---|---|---|
| Actual | 0 | 640708 | 112491 |
| | 1 | 636 | 1009 |

ROC Curve:

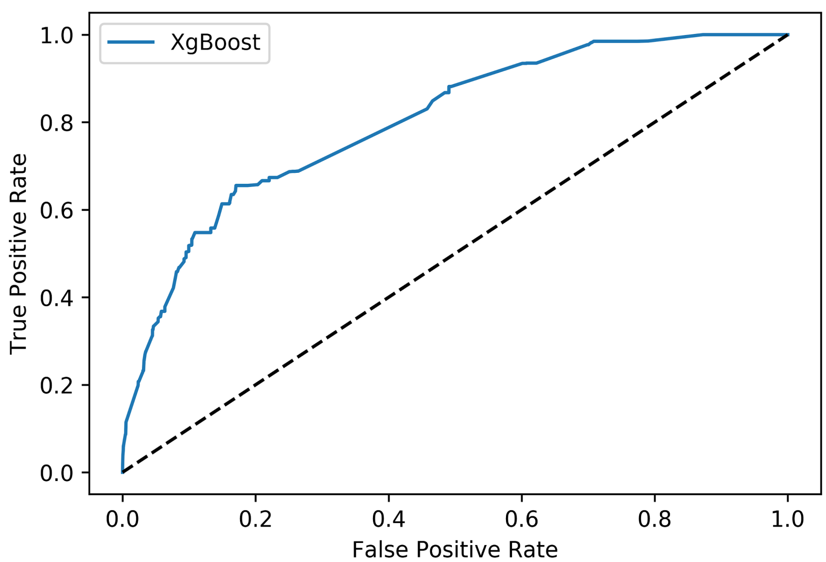


*Fig.4 (e) ROC Curve for XgBoost Classifier with AUC = 0.802*

The results for XgBoost may seem alright looking at the AUC curve but we wouldn't recommend that a data center use this model for predictive maintenance of hard drives because the false positive rate is extremely high. This would mean the operators would have to manually look at over 100,000 hard drives. This is not practical and defeats the purpose of using data mining techniques for ensuring data storage reliability.

**6. KNN Classifier:**

| | |
|---|---|
| Accuracy on Test Data: | 0.99 |
| Recall: | 0.762 |
| Precision: | 0.882 |
| False Positive Rate: | 0.0002 |
| F1 Score: | 0.818 |

| Confusion Matrix | | Predicted | |
|---|---|---|---|
| | | 0 | 1 |
| Actual | 0 | 753031 | 168 |
| | 1 | 391 | 1254 |

ROC Curve:

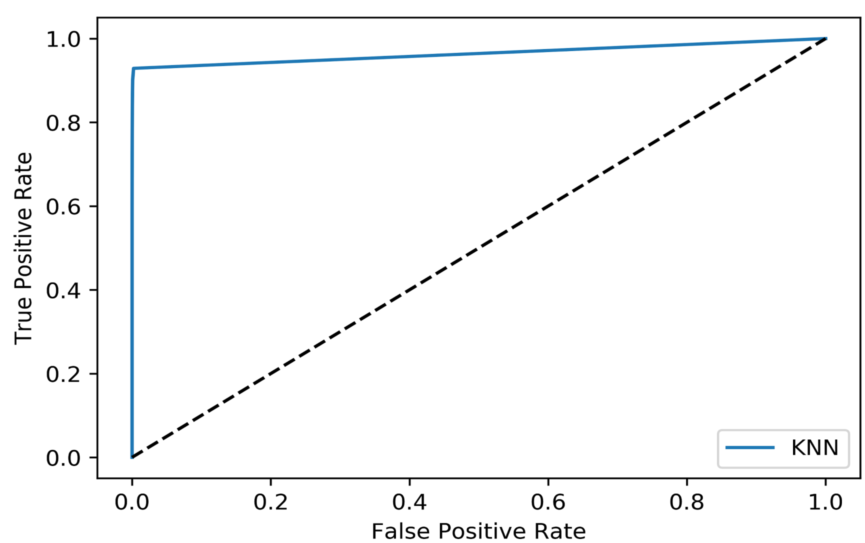


*Fig.4 (f) ROC Curve for KNN Classifier with AUC = 0.964*

The k-Nearest Neighbor classifier (k = 10) also performs well. It reports more false negatives than Decision Tree or the AdaBoosted tree but classifies the unhealthy hard drives correctly decently well.

**Model Selection based on Performance Analysis**

Table 3 consolidates the performance results of the various models built as part of our project. The better performing models are highlighted in green and the poorest model is highlighted in red.

*Table 3. Performance results for models used*

| Model | AUC ROC | Recall | Precision | F1 Score |
|---|---|---|---|---|
| **Logistic Regression** | 0.581 | 1 | 0.00218 | 0.0043 |
| **Decision Tree Classifier** | 0.962 | 0.897 | 0.89 | 0.893 |
| **Adaboost Decision Tree Classifier** | 0.955 | 0.899 | 0.9 | 0.9 |
| **Gradient Boost Classifier** | 0.887 | 0.342 | 0.767 | 0.473 |
| **XgBoost Classifier** | 0.802 | 0.613 | 0.009 | 0.0175 |
| **KNN Classifier** | 0.964 | 0.762 | 0.882 | 0.818 |

# V. EVALUATION

In Chapter 4, we saw the ROC curves for the various models that were built. Table 3 makes it easy to compare and note that Decision Tree and AdaBoosted Decision Tree classifiers have the highest areas under their ROC curves. To reaffirm that these models indeed perform well, we also cross validated the model training to get a generalized overview of their performances. Fig. 3 shows a comparison of the different models used and their 10-fold cross validated AUCs.

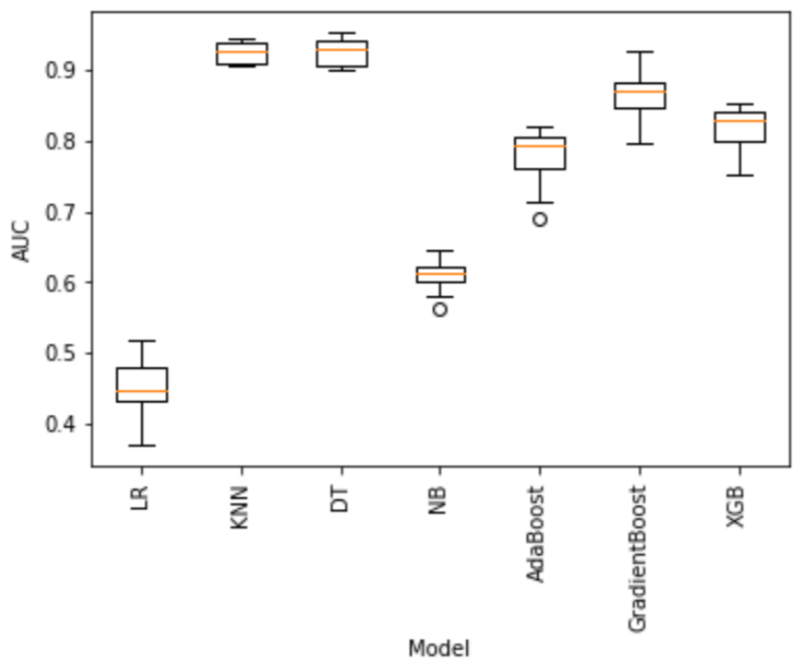


*Fig. 3. Boxplot comparison of the cross validated results for various machine learning models*

In our study, the cost of a false positive is the manual labor involved in checking on a healthy hard drive that has been reported unhealthy and the cost of a false negative is missing hard drive failure (data loss). Since we cannot quantify the exact cost, we are unable to plot profit curves to calculate an exact return on investment.

# VI. DEPLOYMENT

Our Hard Drive Failure prediction model has been trained to predict if a hard disk is going to fail 45 days ahead of time. Hence, our prediction model would need hard drive SMART values of all hard drives in the data center on a daily basis. The same preprocessing steps used while building our model have to be applied to daily reported data.

One way to design this would be to implement a continuous integration and deployment process, which would automatically collect the data from the hard drive logs of the data centre, extract the required SMART values and predict if any of the hard drives show signs of imminent failure.

Currently, our model is designed for a particular make (Seagate) and hard drive model (ST4000DM000) owing to the fact that the SMART values vary across multiple manufacturers and models. In the future, we could expand our study and build models for different manufacturers/models.

As far as application development goes, we would propose to implement the below features:

- To align with a company's Business Continuity Plan, the model would get the real-time SMART stats and feed the predictions to a Data Center. Here the hard drives can be prioritized depending on the number of failure warnings collected over time. The Data Center would also collect historical data which help us to further train the model in the future.
- Build an interactive dashboard, which could display the real-time SMART values for the hard drives according to their priorities. This would give the technicians/operations engineer handling these incidents, some insight to take a decision regarding the future course of action.
- Ability to report if our predictions were accurate (both false alarms and hit rate). This would help in evaluating the real-world performance of our application and would help in building a better model in the future.

One of the possible caveats of our application would be that manufacturers often add new SMART values periodically which can be logged, again our model would not be factoring these new SMART values in the model training. If the new SMART values are predictive of failures, they would need to be analyzed and the model would need to be built again.

# VII. CONCLUSION

In this project, several classification algorithms are modeled on the SMART hard drive dataset from a real-world dataset Backblaze. The fact that hard drive failures being rare events implies that class imbalance in the dataset needs to be addressed. A good classifier should be able to distinguish the two classes in this skewed dataset. We selected performance metrics based on true positive/false positive rates and tested several algorithms. It was found that Decision Tree, AdaBoosted Decision Tree and k-Nearest Neighbor models have high recall and low false positive rates. Given that all three classifiers have comparable AUCs, the AdaBoosted classifier having the highest recall percentage and lowest false positive rate can be concluded to be the best performing model.

# VIII. REFERENCES

[1] Backblaze, “Hard Drive Data and Stats,” Backblaze, [Online]. Available: https://www.backblaze.com/b2/hard-drive-test-data.html.

[2] N. Aussel, S. Jaulin, G. Gandon, Y. Peteti, E. Fazli and S. Chabridon, “Predictive Models of Hard Drive Failures based on Operational Data,” in *16th IEEE International Conference on Machine Learning and Applications*, Wessling, Germany, 2017.

[3] A. Klein, “What SMART Hard Disk Errors Actually Tell Us. , ,,” 6 Oct 2016. [Online]. Available: www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures/.

[4] B. Beach, “Backblaze Blog | Cloud Storage & Cloud Backup,” backblaze, 27 Jan 2015. [Online]. Available: www.backblaze.com/blog/hard-drive-smart-stats/. [Accessed 10 March 2019].

[5] T. Pitakrat, A. v. Hoorn and G. Lars, “A comparison of machine learning algorithms for proactive hard disk drive failure detection,” in *Proceedings of the 4th international ACM Sigsoft symposium on Architecting critical systems*, ACM, 2016.

# IX. CONTRIBUTIONS

All of us initially discussed about the possible approaches to solving this business problem based on the topics discussed in the class. Each of us did research individually to check the current studies that have been performed in the area of predictive maintenance.

Contributions by each member are more specifically listed below:

- **Tushar Bhatkal** suggested and implemented different machine learning models which we could use for highly unbalanced data. Additionally, he worked on the initial feature analysis and feature engineering.
- **Satyajeet Maharana** analyzed and suggested the dataset features to use and worked on the initial loading and cleaning up of data using Spark and HDFS. He also analyzed the specific manufacturers and their statistics in the dataset to suggest the ideal make and model of hard disks we have used.
- **Varsha Murali** researched and implemented different approaches to find a suitable solution to our business problem. She suggested the approach used in our study where we changed the ‘failure’ column for each failing hard drive for 45 days prior to failure date during the data modeling phase.

We collaborated on developing code to perform data analysis and modeling using Google Colab notebooks with access to the dataset via a shared drive folder. Each of us worked on the training and testing of all the models and compared results in the end. We also worked on writing the report together.