3/12/2018

AN EMPIRICAL STUDY ON THE TEMPORAL DEPENDENCE OF S.M.A.R.T. METRICS IN PREDICTING HARD DRIVE FAILURES

SUBMITTED TO

Dr. Fan ZhenZhen & Dr. Zhu Fangming

INSITITUTE OF SYSTEMS SCIENCE

NATIONAL UNIVERSITY OF SINGAPORE

PREPARED BY

GOPALAKRISHNAN SAISUBRAMANIAM(A0178249N)

MADAN KUMAR MANJUNATH(A0178237W)

GANANATHAN KHOTEESWARUN(A0178328U)

CHOKKALINGAM SHANMUGASIVA(A0178230J)

ANJALI SINHA(A0178476L)

KENNETH RITHVIK(A0178448M)

MASTER OF TECHNOLOGY IN KNOWLEDGE ENGINEERING

BATCH KE-30(2018)

Table of Contents

1. Business Understanding

1.1. Business Objective

Data centers generally use hard drives as data storage device. Large companies heavily rely on data and use many hard drives, which become challenging to monitor manually. When there is an issue with the hard-disk, it should function for at least next 24 hours for the data back-up to be done. But in ideal cases, the hard-disk fails even before 24 hours resulting in loss of data. Hard drive failures cause data loss which can cause a serious problem for the users. As backup, multiple copies of data can be stored in the system, but it might increase the cost at the same time. In industries, hard drives are monitored by setting threshold for several critical metrics.

1.2. Assess Situation

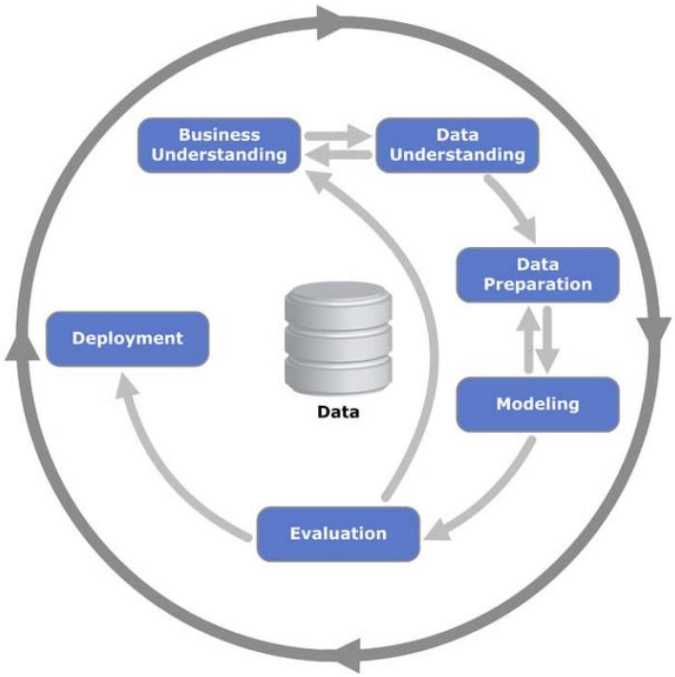
SMART (Self-Monitoring, Analysis, and Reporting Technology) attributes of hard disks can be useful in detecting the failure rate of the hard-disks. It is useful to predict the hard drive failure by developing a model, so that it can be used to get useful insights to improve the system reliability and help cut cost.

1.3. Data Mining Goals

A single hard disk may have up to 30 SMART attributes which may give health information of the hard disk and provide statistical information about the hard disk. There are machine learning and statistical methods available to predict hard disk failure based on SMART attributes.

1.4. Project Plan

We will be following the CRISP-DM Model for achieving the business and data mining goals for this project.



[Figure 1 – CRISP-DM methodology]

The following steps have been carried out in order to discover knowledge from the data:

- Business Understanding
- Data Understanding

- Data Preparation
- Modeling
- Evaluation

2. DATASET

2.1. COLLECT INITIAL DATA

The dataset under consideration is hard drive dataset, published by Backblaze[1]. Backblaze records SMART stats of 67,814 hard drives, which are running every day in their Sacramento data center. SMART stands for Self-Monitoring, Analysis and Reporting Technology, is a monitoring system included in hard drives to report attributes about a given drive.

2.2. DESCRIBE DATA

Each day a snapshot of each operational hard drive is taken in the Backblaze data center. The snapshot will have the basic drive information along with the SMART statistics reported by that drive. The daily snapshot of one drive is one record or row of data. The snapshots of the drives are compiled into a single file which further consists of separate row to which denotes the status of the hard drive. The detailed description of dataset is as follows.

Index	Column #	Variable Name	Type	Description
A	1	Date	Date	Date (file created) in YYYY-MM-DD format.
B	2	Serial Number	Categorical	Serial number of the drive assigned by the manufacturer.
C	3	Model	Categorical	Model number of the drive assigned by the manufacturer.
D	4	Capacity	Numeric	The capacity of the drive (in bytes)
E	5	Failure	Categorical	"0" – Drive is working fine. "1" – if this is the last day the drive was working fine before failure.
F	6-95	SMART Stats	Numeric	Stats of the drive available in 90 columns which are further split into Raw and Normalized values for 45 different SMART stats.

[Table 1: Data Description]

2.3. EXPLORE DATA

The full version of the dataset comprises of day-wise observations covering ~67,814 hard drives over the span of Jan 2015 – Dec 2017. (data before 2015 is not available) The figure below provides statistics on the failure rates for each model of hard drive being used.

Hard Drive Annualized Failure Rates

Reporting period April 2013 - December 2017 inclusive

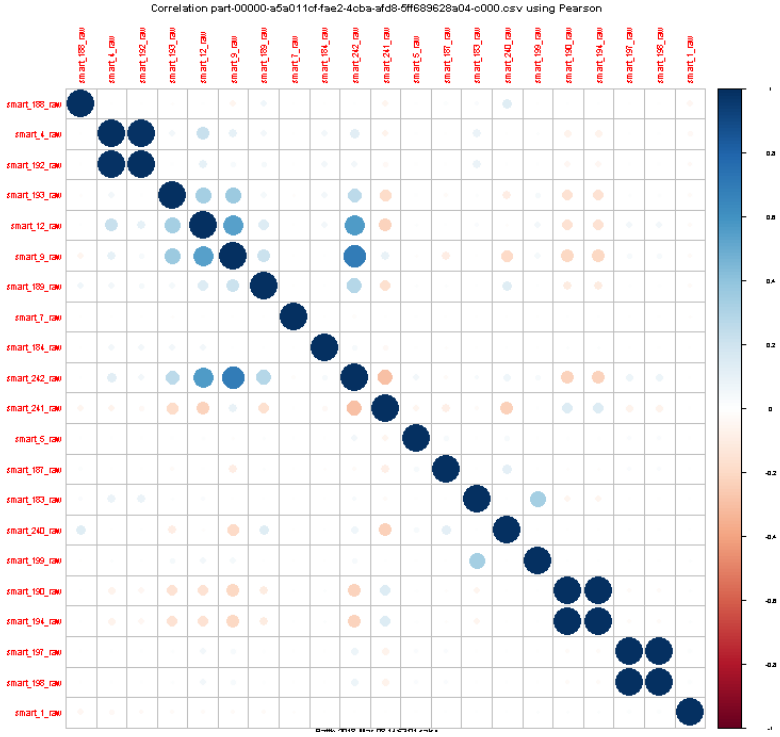
MFG	Model	Drive Size	Drive Days	Drive Failures	Annualized Failure Rate	Confidence Interval Low	Confidence Interval High
Seagate	ST12000NM0007	12 TB	294,924	17	2.10%	1.2%	3.4%
Seagate	ST10000NM0086	10 TB	121,017	3	0.90%	0.1%	2.2%
HGST	HUH728080ALE600	8 TB	47,483	2	1.54%	0.2%	4.5%
Seagate	ST8000DM002	8 TB	4,579,774	138	1.10%	0.9%	1.3%
Seagate	ST8000NM0055	8 TB	2,636,788	89	1.23%	1.0%	1.5%
Seagate	ST6000DX000	6 TB	1,882,520	57	1.11%	0.8%	1.4%
WDC	WD60EFRX	6 TB	499,533	62	4.53%	3.5%	5.8%
Toshiba	MD04ABA500V	5 TB	46,170	2	1.58%	0.2%	5.7%
HGST	HMS5C4040BLE640	4 TB	9,394,769	136	0.53%	0.5%	0.6%
HGST	HDS5C4040ALE630	4 TB	4,280,569	93	0.79%	0.6%	1.0%
Seagate	ST4000DM000	4 TB	35,168,535	2,850	2.96%	2.9%	3.1%
Seagate	ST4000DM001	4 TB	78,503	33	15.34%	10.6%	21.5%
Seagate	ST4000DM005	4 TB	1,255	1	29.08%	0.0%	120.0%
Toshiba	MD04ABA400V	4 TB	141,381	4	1.03%	0.3%	2.6%
WDC	WD40EFRX	4 TB	63,127	4	2.31%	0.6%	5.9%
HGST	HMS5C4040ALE640	4 TB	8,797,680	130	0.54%	0.5%	0.6%
WDC	WD30EFRX	3 TB	1,233,586	171	5.06%	4.3%	5.9%
		Totals	**69,267,614**	**3,792**	**2.00%**		

BACKBLAZE

[Figure 2: Backblaze Hard Drive Failure Rates [1]]

We consider taking Seagate model number *ST4000DM000* as a subset for the analysis. This subset shall be henceforth referred to simply as the *dataset* in this document.

The following is the correlation matrix generated using the SMART raw values:

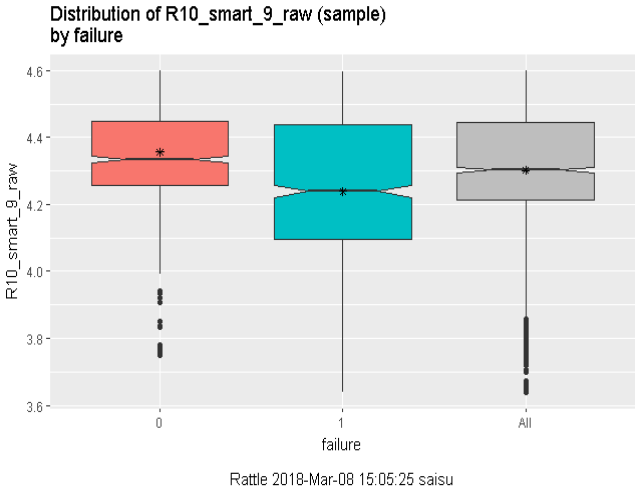


[Figure 3: Initial Exploration - Correlation Matrix]

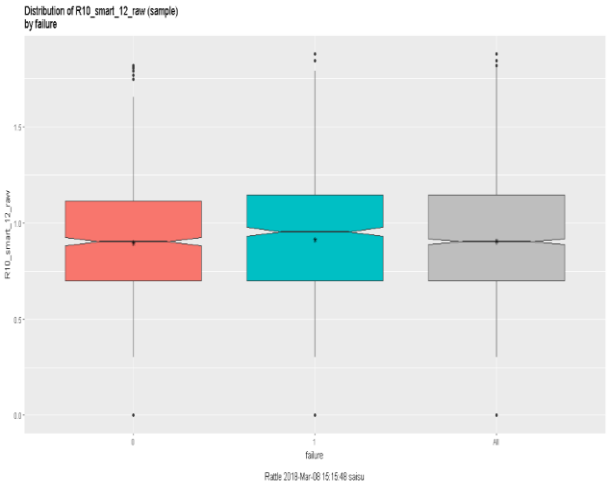
Inferences drawn from the correlation matrix are as follows:

- SMART 4 and 192 exhibit high correlation as they relate to the number of cycles on start after shutdown. 192 captures power off cycles and is complemented by 4 which increments the value on startup.
- SMART 190 and 194 deal with temperature, hence highly correlated.
- SMART 197 and 198 exhibit high correlation because 197 defines unstable sectors due to read errors and 198 gives count of uncorrectable errors while read/write to a sector. We take 198 and ignore 197.
- SMART 9,12 and 242 are correlated to an extent as they cover related features - number of hours the drive is up, count of full power on/off cycles, and the Logical Block Addresses read during the time it was up.

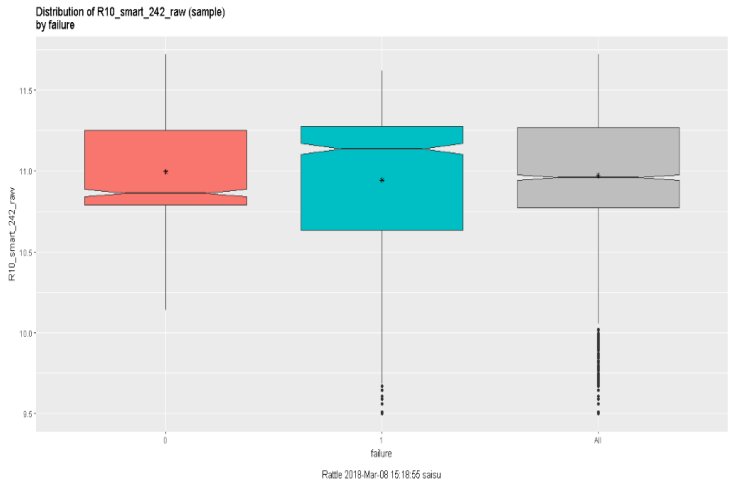
Our selection of the hard drives was based on SMART 9, to analyze hard drives that were running for the past 6 months or more. Since SMART 9, 12 and 242 are correlated the distribution is almost equal as seen in the box plot below (Fig 4, Fig 5 and Fig 6]. We ignore the three SMART readings based on the above reasons.



[Figure 4: SMART 9 distribution]



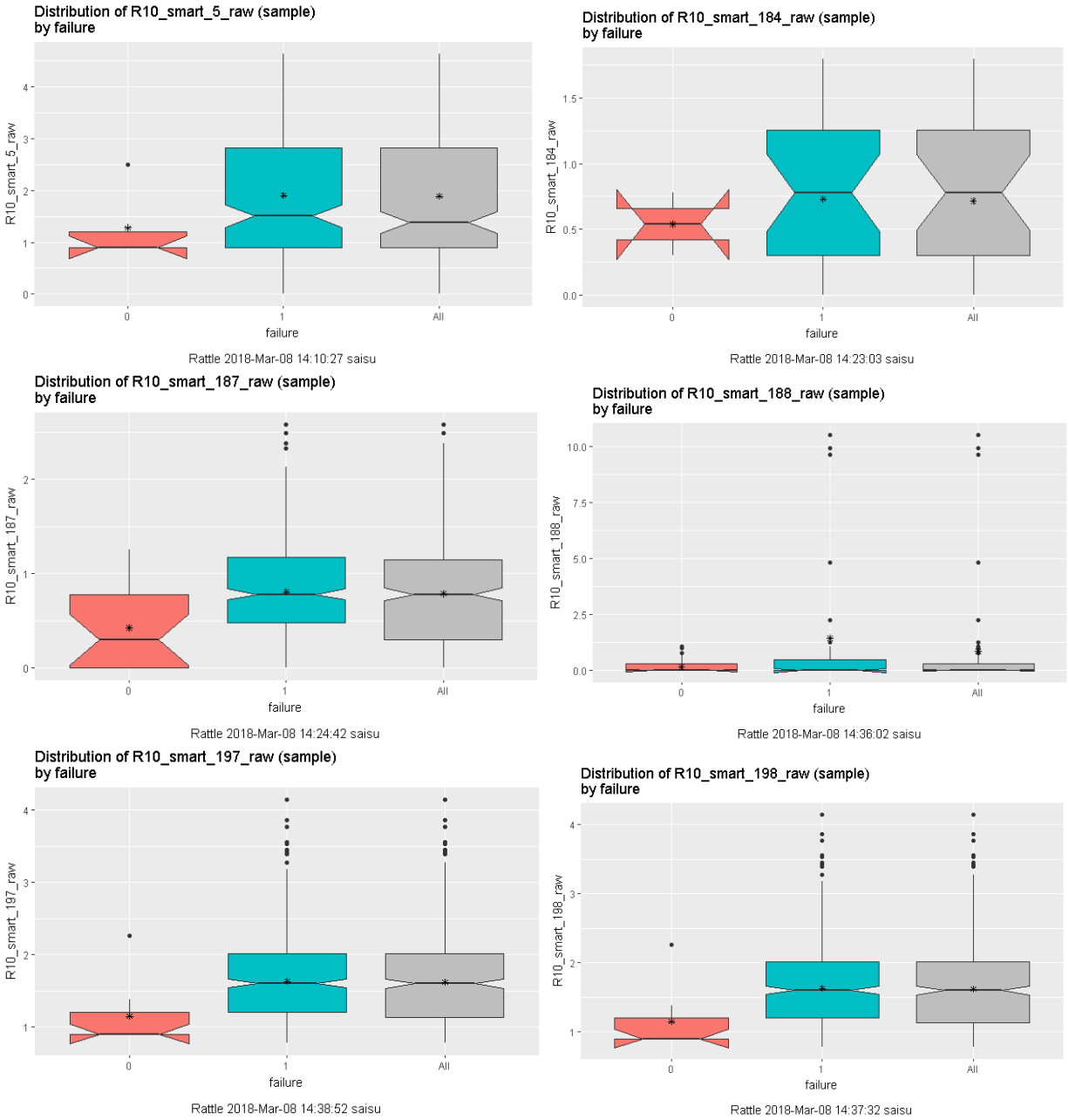
[Figure 5: SMART 12 distribution]



[Figure 6: SMART 242 distribution]

Temperature is not a good feature for our analysis since there are no incidents of extreme overheating, hence we also omit 190 and 194. SMART 4 and 192 do not influence the target failure as much as the below main influencers.

The below SMART ids exhibited variance in the data distribution:



[Figure 7: Box Plot Comparison of SMART Ids]

They are high influencers of drive failures and are considered as part of the critical features

SMART ID	Attribute Name	Description	Comments
05	Count of re-allocated sectors	The raw value of the no. of bad sectors that were found and remapped.	This metric has been used to indicate the life expectancy of the drive.
07	Seek error rate	The raw value gives the drive's magnetic head seek error rate.	Different value measurements reported by different vendors.

			For the same vendor and model, the values should be consistent.
183	Runtime bad block	The total no. of data blocks with detected and un-correctable errors occurred during regular operations.	An indicator of drive aging and/or potential electromechanical problems
184	End to end error	Contains the parity error count that exists in the data path to the media through the drive's cache RAM.	Ideal value should be low as parity errors occur when data gets corrupted during transmission.
187	Un-correctable errors	No. of errors that could not be corrected using hardware ECC.	High error count is a sign of failing drive.
188	Aborted operations	The no. of operations aborted due to hard disk drive timeout.	This value is close to 0 for healthy drives.
197	Pending sector count	The no. of unstable sectors that are to be remapped due to un-recoverable read errors.	The sector is remapped, and this value is decreased over time on subsequent successful reads.
198	Count of un-correctable errors	Total no. of un-correctable errors occurred while reading or writing errors.	A rise in the value of this attribute indicates defects of the disk surface and/or problems in the mechanical subsystem.

[Table 2: Hard Drive SMART Matrices]

These SMART ids are also acknowledged in the Backblaze website to be relevant for further feature extraction. Though SMART 7 (Seek error rate) is not preferred as part of the list owing to different value measurements by different vendors, we take it into consideration since we focus on just one vendor and model.

2.4. Verify Data Quality

There are some SMART features whose readings have not been recorded, so these come up as blanks/NaN in the dataset. Also, there are some features for which the values for all records are constant. To avoid inconsistencies in modelling and analysis, we are considering the raw values of only Seagate model number ST4000DM000. The is no noise in the dataset, as the readings were taken directly from hard disks and there was no external interference.

3. Hypothesis

Time series analysis has been proven to yield better results than the traditional i.i.d (independent and identically distributed) assumption for most real-world use cases. We believe considering the temporal dependence of SMART readings will improve the sensitivity and thereby the model performance. We compare the results without and with temporal dependency assumption. For the latter, we consider time ranges of 3, 7, 15 and 30 days and compare the model performance for different train and test splits.

4. Data Preparation

4.1 Data Selection

The original dataset is of considerable volume and to lower our scope, we selected a Seagate model and filtered by drives older than 6 months in operation. After crunching down 3 years' worth of data, we were able to obtain 1306 cases of reported failures. We took an equivalent amount of drives that are still operational, 1527 in number.

The following were the steps involved

- SparkR was chosen to reduce the size of dataset as traditional R cannot scale to such volume (3GB) and due to time constraints.
- The list of all failed devices was first filtered using combination of SparkR and SparkSQL. SparkSQL was opted than pure native R method because it has been observed that it is often more efficient to work with data with an indexed field (serial_number) for such large-scale computations using some form of SQL query [4].
- A new column lifetime was created as we take the drive age to be greater than 6 months and the dataset was filtered using the same.
- The data was partitioned by serial_number and sorted descending by date to enable us to get multiple datasets based on the window interval (for below experiments)

4.2 Data Cleaning

The SMART ids that were not recorded for ST4000DM000 model number were removed from the dataset containing as the values contained blanks/NaN or constant values.

4.3 Data Construction & Transformation

4.3.1 Data without Temporal consideration

We consider a portion of the dataset focusing on the last date of hard drive operation, i.e. a set of records comprising of metrics with the target '1' indicating failure by end of day and an equivalent set of records comprising of metrics with the target '0' indicating they are still operational.

4.3.2 Data With Temporal Consideration

Time series/Temporal data is not ready to use format for most machine learning algorithms. The sequential data should be modelled according to the algorithm considered and business case.

The two most popular ways to transform time series data to make it suitable for classification are:

1) Applying a rolling time window
2) Extract features from a temporal sampling frame
   Types of features:
   - Date/Time Feature
   - Lag Features
   - Window Features

However, before transforming the data using these two techniques we will use the last day SMART metrics to predict the failure. Also, apply an exponential moving window to smooth the data and use the last day SMART value to predict the failure. These two models will be used as base models. The above-mentioned techniques will be used on top of these to build and compare the increase in accuracy and sensitivity of our prediction.

Exponential smoothing:

Smoothing is a statistical process where a moving average is used to transform data. By doing so we will be eliminating random variations(noise) from our temporal data. An improved form of simple smoothing is weighted moving average, in this method the most recent time interval preceding the prediction time frame is given the most significant is calculating the smoothed value. In exponential smoothing we provide the smoothing constant which is used to calculate the weights of each interval, and thus does not require us to provide to provide the weights explicitly.

Tsfresh Package:

Tsfresh is a python library that was specifically developed to extract meaningful and relevant features from time series data. The features that are extracted, can then be used further for model building tasks like classification, regression or clustering. Tsfresh automates the whole-time series feature extraction process, that would otherwise have had to be done manually. This library calculates around 790 features for each time series metric supplied to it. When supplied with the target variable, the library automatically computes which features are relevant to the target variable and which are not, thereby removing the un-necessary features that are computed.

We generated 790 new features for each of the features above, which were filtered down to 390 relevant features by *tsfresh* based on the FDR(False Discovery Rate) [3] value. (threshold=0.05)

Some of the important features that were extracted for each metric are as follows:

Extracted Feature	Description
abs_energy(metric)	Denotes the absolute energy of the smart metric which is the cumulative result of the squared values
agg_autocorrelation(metric, param)	Denotes the result returned by the aggregation function
autocorrelation(metric, lag)	Denotes the autocorrelation of the specified lag
count_below_mean(metric)	Denotes the no. of values in the smart metric that are below the average of the smart metric readings
skewness(metric)	Denotes the skewness of the metric (calculated by adjusted Fisher-Pearson standardized moment coefficient G1)
fft_coefficient(metric, param)	Denotes the Fourier coefficients of the one-dimensional discrete Fourier Transform for real input
kurtosis(metric)	Denotes the kurtosis of the metric(calculated by adjusted Fisher-Pearson standardized moment coefficient G2).

[Table 3: Tsfresh Selected Features]

We then did Principal Component Analysis to eliminate the redundant SMART features, which helped us further in predicting the failure rate of the hard disks. By transforming the features linearly, principal components were obtained thereby eliminating the multi collinearity problem among the SMART features. Eigen values and Eigen vectors were computed to measure the magnitude and the direction of variation retained by each Principal component. Examining the values of Eigen values helped us in determining the number of principal components which can be considered.

5. MODELLING

5.1 SELECT MODELLING TECHNIQUE

The following modeling techniques were used to predict the hard disk failure using the extracted features:

- Decision Tree

- SVM
- Naive Bayes
- Random Forest
- GLM(Logistic)
- XG boost

Few features we considered before using the listed models:

- In our given problem the decision tree performs equally well in terms of stability and in performance compared to other state of the art models. Additionally, it is easy to interpret and straightforward to visualize, this will help us to explain it to business.
- One important caveat in using the decision tree model is that it tries to make an optimal prediction at every node level. This makes it prone to over fitting, especially when it is deep. This is due to the amount of specificity at each node level. To avoid this pitfall, we build a random forest to compare with it.
- After a thorough cleaning and applying principal component analysis, the data set was apt for applying support vector model for classifying the data. The SVM model gave satisfactory result without the need to apply Kernel method.
- In a GLM (Generalized linear model) the distribution family is decided to depend on the nature of target which has to be targeted. In our case, the target is a binary output which says if the given hard disk failed or not on that day. Thus, we choose the logistic model (Binomial family)

5.2 Generate Test Design

While designing the tests that need to be carried out on the model that was built, we decided to carry empirical validation. In this process the dataset was split using the simple split technique. We use multiple split dimensions on each of our different modelling techniques to find the accuracy in every scenario.

The split sizes that were used were: -

- 50/50
- 70/30
- 80/20

We also tried building our model with data from different intervals of time that we choose, like 1, 3, 7, 15, 30 days.

5.3 Build Model

The dataset was split into train and test sets based on the test design. For each trial, the records were sampled using stratified random sampling and trained on each of the selected model. 5-fold cross validation was the evaluation criteria before the final test prediction. PCA was used as the dimensionality reduction technique and the Principal Components were generated for each validation set using the coefficients in each of the cross-validation fold. The same approach was also considered for the final test sets.

The sample runs can be traced to the Appendix section. The experiments were performed in R using the caret package.

5.4 Assess Model

Depending on the type of output the model creates we assess them differently:

Class output: Models which outputs either of the binary as result ie: SVM etc

Probability output: Provides its confidence for each class ie: Random Forest etc

As assessing a model is an integral part of model building these parameters were taken into consideration:

Confusion Matrix:

It is an N*N matrix, where N denotes no of classes predicted.

From the confusion matrix we derive these **sub-metrics**:

Accuracy - % of correct predictions

Kappa - a metric that compares an Observed Accuracy with an Expected Accuracy

Precision - % of positive cases which were correct

Sensitivity/Recall - a portion of actual positive cases which were predicted correctly

Specificity - a portion of actual negative cases which were predicted correctly.

ROC and Area under the Curve - A plot of TRP vs FPR to showcase the strength of the model.

6. EVALUATION

6.1 EVALUATE RESULTS

The models built using the last day reading was taken as the baseline. Then feature extraction was applied and feature engineering on the temporal aspects of the variables generated new features. These features were used to train different models and evaluate them using the validation and test sets.

6.1.1 BASELINE RESULTS

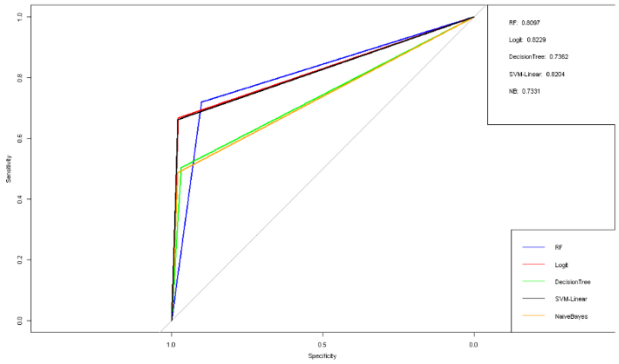
Classifier	Train/Test Split	Interval (days)	Fold (k)	Accuracy	Kappa
Decision Tree (CART)	80/20	1	K=5	0.8226751	0.6341372
	70/30	1	K=5	**0.8276226**	0.6447503
	50/50	1	K=5	0.8214744	0.6324752
	80/20	1	K=LOOCV	0.8231142	0.6351325
SVM - Linear	80/20	1	K=5	0.8306046	0.6510718
	70/30	1	K=5	**0.8377045**	0.6660676
	50/50	1	K=5	0.8284541	0.6462033
	80/20	1	K=LOOCV	0.8345831	0.6594188
Naive Bayes	80/20	1	K=5	0.8195972	0.6285382
	70/30	1	K=5	0.8250126	0.6405858
	50/50	1	K=5	0.7403113	0.4563025
	80/20	1	K=LOOCV	**0.8253198**	0.6405247
Random Forest(t=51)	80/20	1	K=5	0.8345830	0.6620415
	70/30	1	K=5	0.8402336	0.6739337
	50/50	1	K=5	0.8320485	0.6575127
	80/20	1	K=LOOCV	**0.8407587**	0.6725054
GLM	80/20	1	K=5	0.8306065	0.6510585
	70/30	1	K=5	**0.8356868**	0.6619712
	50/50	1	K=5	0.8278073	0.6450155
	80/20	1	K=LOOCV	0.8345831	0.6594188

[Table 4: Cross Validation results without temporal dependence]

Classifier	Train/Test Split	Interval (days)	Accuracy	Sensitivity	Specificity
Decision Tree (CART)	80/20	1	0.8076	0.6284	**0.9869**
	70/30	1	0.7965	0.6061	**0.9869**
	50/50	1	**0.8100**	**0.6371**	0.9830
SVM - Linear Kernel	80/20	1	**0.8251**	**0.6667**	0.9836
	70/30	1	0.8108	0.6368	**0.9847**
	50/50	1	0.8211	0.6631	0.9790
Naive Bayes	80/20	1	0.8131	0.6590	0.9672

	70/30	1	0.7938	0.6292	0.9585
	50/50	1	0.7165	0.4395	0.9934
Random Forest(t=51)	80/20	1	**0.8364**	0.7318	**0.9410**
	70/30	1	0.8198	0.7008	0.9389
	50/50	1	0.8328	**0.7443**	0.9214
GLM	80/20	1	**0.8268**	0.6667	**0.9869**
	70/30	1	0.8033	0.6240	0.9825
	50/50	1	0.8255	**0.6708**	0.9803

[Table 5: Test results without temporal dependence]



[Figure 8: ROC without temporal dependence]

6.1.2 Exponential Moving Average Results

Classifier	Train/Test Split	Interval (days)	Accuracy	Sensitivity	Specificity
Decision Tree (CART)	80/20	3	**0.8350**	0.6897	0.9803
SVM - Linear	80/20	3	0.7053	0.4138	0.9967
Naive Bayes	80/20	3	0.6609	0.3218	**1.0000**
Random Forest(t=51)	80/20	3	0.7823	**0.7778**	0.7869
GLM	80/20	3	0.8112	0.6322	0.9902
Decision Tree (CART)	80/20	7	**0.8509**	0.7280	0.9738
SVM - Linear	80/20	7	0.7116	0.4330	**0.9902**
Naive Bayes	80/20	7	0.7009	0.4215	0.9803
Random Forest(t=51)	80/20	7	0.7996	**0.7893**	0.8098
GLM	80/20	7	0.8292	0.6782	0.9803
Decision Tree (CART)	80/20	15	**0.8268**	0.6897	0.9639
SVM - Linear	80/20	15	0.7332	0.4828	0.9836
Naive Bayes	80/20	15	0.6617	0.3333	**0.9902**
Random Forest(t=51)	80/20	15	0.7845	**0.7625**	0.8066
GLM	80/20	15	0.8161	0.6552	0.9770
Decision Tree (CART)	80/20	30	**0.8273**	0.6973	0.9574
SVM - Linear	80/20	30	0.7392	0.4981	**0.9803**
Naive Bayes	80/20	30	0.6686	0.3602	0.9770
Random Forest(t=51)	80/20	30	0.7996	**0.7893**	0.8098
GLM	80/20	30	0.8186	0.6667	0.9705

[Table 6: Test results with Exponential Smoothing Average]

6.1.3 FEATURE ENGINEERING USING TSFRESH RESULTS

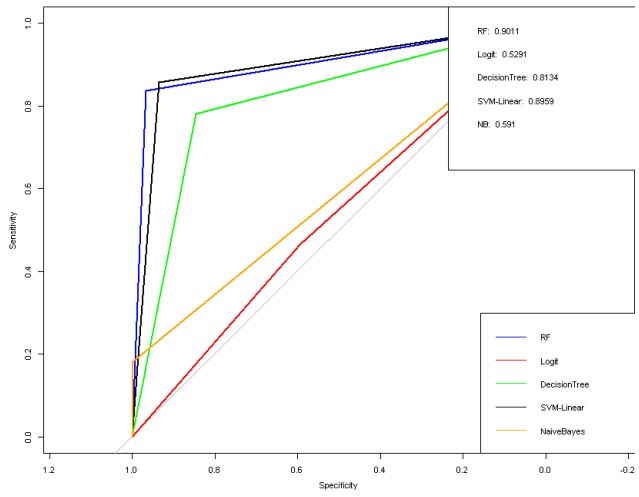
Classifier	Train/Test Split	Interval (days)	Fold (k)	Accuracy	Kappa
Decision Tree (CART)	80/20	30	K=5	0.8773706	0.7556865
SVM - Linear	80/20	30	K=5	0.8989799	0.7957245
Naive Bayes	80/20	30	K=5	0.6228443	0.1932852
Random Forest(t=51)	80/20	30	K=5	0.9461787	0.8913123
GLM	80/20	30	K=5	0.5342397	0.07688918

[Table 7: Cross Validation results with temporal dependence without dimensionality reduction]

Below we tabulate results based on the new features without any dimensionality reduction.

Classifier	Train/Test Split	Interval (days)	Accuracy	Sensitivity	Specificity
Decision Tree (CART)	80/20	30	0.8551	0.9464	0.7639
SVM - Linear	80/20	30	0.8977	0.8774	0.9180
Naive Bayes	80/20	30	0.59962	0.19923	1.0000
Random Forest(t=51)	80/20	30	0.9478	0.9349	0.9607
GLM	80/20	30	0.4789	0.9579	0.0000

[Table 8: Test results with temporal dependence without dimensionality reduction]



[Figure 9: ROC with temporal dependence without dimensionality reduction, 30-day interval]

Since 790 features increase time complexity of training each model, we reduced the dimensions using PCA. Below are the results after dimensionality reduction to 390 features.

Classifier	Train/Test Split	Interval (days)	Fold (k)	Accuracy	Kappa
Decision Tree (CART)	80/20	3	K=5	0.8985423	0.7964266
	70/30	3	K=5	0.8921278	0.7834238
	50/50	3	K=5	**0.9029699**	0.8055754
	80/20	7	K=5	0.8725229	0.7445786
	70/30	7	K=5	0.8714755	0.7416908
	50/50	7	K=5	0.8774355	0.7551319

	80/20	15	K=5	0.8489804	0.6936425
	70/30	15	K=5	0.8785309	0.7566148
	50/50	15	K=5	0.8599139	0.7161298
	80/20	30	K=5	0.8491321	0.6919525
	70/30	30	K=5	0.8513065	0.6977534
	50/50	30	K=5	0.8518284	0.6979864
SVM - Linear	80/20	3	K=5	0.9069347	0.8133869
	70/30	3	K=5	0.9072666	0.8135116
	50/50	3	K=5	0.7955951	0.587527
	80/20	7	K=5	0.9104618	0.8199203
	70/30	7	K=5	0.9097779	0.818549
	50/50	7	K=5	0.8086075	0.6241818
	80/20	15	K=5	0.9086919	0.8158689
	70/30	15	K=5	**0.9148271**	0.8281026
	50/50	15	K=5	0.913905	0.8261743
	80/20	30	K=5	0.9073577	0.8124444
	70/30	30	K=5	0.9012009	0.8002441
	50/50	30	K=5	0.8969741	0.7914255
Naive Bayes	80/20	3	K=5	0.6453278	0.2443676
	70/30	3	K=5	0.6385912	0.2290199
	50/50	3	K=5	0.581511	0.0985826
	80/20	7	K=5	0.6545954	0.2651915
	70/30	7	K=5	0.6259955	0.2010134
	50/50	7	K=5	0.6048208	0.1515207
	80/20	15	K=5	0.6453647	0.2444154
	70/30	15	K=5	**0.6764229**	0.3138534
	50/50	15	K=5	0.673976	0.3079158
	80/20	30	K=5	0.6757923	0.3123601
	70/30	30	K=5	0.6562603	0.2693993
	50/50	30	K=5	0.6669346	0.2925059
Random Forest(t=51)	80/20	3	K=5	0.9356002	0.8703924
	70/30	3	K=5	0.9395135	0.8780854
	50/50	3	K=5	0.9343823	0.8679335
	80/20	7	K=5	0.9426632	0.8845685
	70/30	7	K=5	0.9395275	0.8781434
	50/50	7	K=5	0.9385977	0.8764434
	80/20	15	K=5	0.9435268	0.8863029
	70/30	15	K=5	0.9380022	0.8748314
	50/50	15	K=5	0.9442318	0.8877030
	80/20	30	K=5	0.9510381	0.9012101
	70/30	30	K=5	**0.9521131**	0.9033699
	50/50	30	K=5	0.9421588	0.8834348
GLM	80/20	3	K=5	0.5429365	0.08074906
	70/30	3	K=5	0.4668944	-0.01205145
	50/50	3	K=5	0.394216	-0.2134292
	80/20	7	K=5	0.4918293	0.009717047
	70/30	7	K=5	0.6835153	0.3481236
	50/50	7	K=5	0.4528916	-0.08182444
	80/20	15	K=5	0.4429734	-0.1222903
	70/30	15	K=5	0.4651124	-0.09582624
	50/50	15	K=5	**0.5552207**	0.1174163
	80/20	30	K=5	0.4577219	-0.05796136
	70/30	30	K=5	0.4539399	-0.06879593

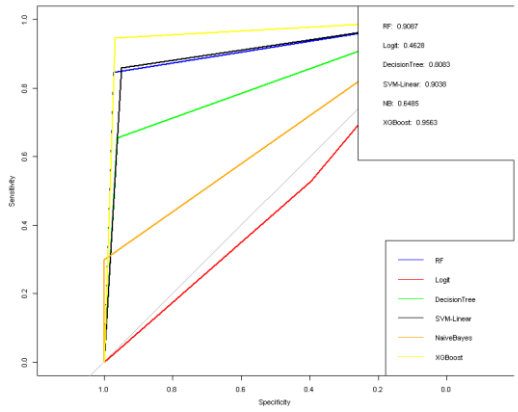
	50/50	30	K=5	0.4869659	-0.0288373
XGBoost	80/20	3	K=5	0.9457401	0.8907149
	70/30	3	K=5	0.9440513	0.8872984
	50/50	3	K=5	0.9421263	0.8836066
	80/20	7	K=5	0.9501532	0.8995988
	70/30	7	K=5	0.9521207	0.9034856
	50/50	7	K=5	0.9435347	0.8863664
	80/20	15	K=5	0.9541238	0.9075541
	70/30	15	K=5	0.9445488	0.8882761
	50/50	15	K=5	0.9449460	0.8891294
	80/20	30	K=5	**0.9602980**	0.9199823
	70/30	30	K=5	0.9541256	0.9076314
	50/50	30	K=5	0.9527124	0.9046538

[Table 9: Cross Validation results with temporal dependence]

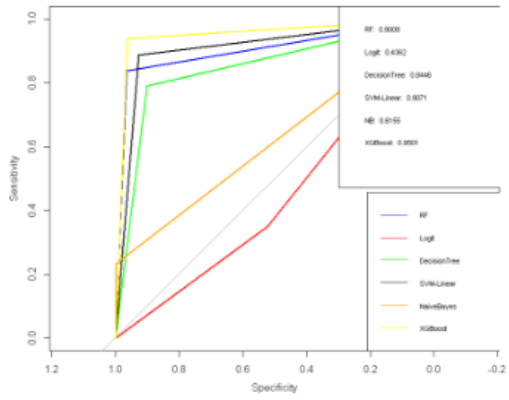
Classifier	Train/Test Split	Interval (days)	Accuracy	Sensitivity	Specificity
Decision Tree (CART)	80/20	3	0.9248	**0.9579**	0.8918
	70/30	3	**0.9069**	0.9207	0.8930
	50/50	3	0.9055	0.9250	0.8860
	80/20	7	0.8617	0.9004	0.8230
	70/30	7	0.9021	0.9156	0.8886
	50/50	7	0.8908	0.8943	0.8873
	80/20	15	0.8873	0.8927	0.8820
	70/30	15	0.8917	0.8772	0.9061
	50/50	15	0.8641	0.8239	0.9043
	80/20	30	0.8619	0.8123	0.9115
	70/30	30	0.8593	0.7928	**0.9258**
	50/50	30	0.8586	0.7933	0.9240
SVM - Linear	80/20	3	**0.8922**	**0.8697**	0.9148
	70/30	3	0.8741	0.8005	0.9476
	50/50	3	0.8828	0.8469	0.9187
	80/20	7	0.8148	0.6820	0.9475
	70/30	7	0.8397	0.7033	0.9760
	50/50	7	0.8242	0.7152	0.9332
	80/20	15	0.8229	0.6590	**0.9869**
	70/30	15	0.8313	0.6931	0.9694
	50/50	15	0.8202	0.6692	0.9712
	80/20	30	0.8251	0.6667	0.9836
	70/30	30	0.8146	0.6445	0.9847
	50/50	30	0.8201	0.6585	0.6585
Naive Bayes	80/20	3	**0.8558**	**0.7510**	0.9607
	70/30	3	0.8207	0.6675	**0.9738**
	50/50	3	0.8312	0.6953	0.9672
	80/20	7	0.8290	0.6973	0.9607
	70/30	7	0.8170	0.6777	0.9563
	50/50	7	0.8136	0.6784	0.9489
	80/20	15	0.8279	0.6820	0.9738
	70/30	15	0.8370	0.7263	0.9476
	50/50	15	0.8150	0.6876	0.9423
	80/20	30	0.8260	0.6782	0.9738
	70/30	30	0.8194	0.6650	0.9738
	50/50	30	0.8141	0.6478	0.9803

Random Forest(t=51)	80/20	3	**0.9327**	**0.9540**	0.9115
	70/30	3	0.9135	0.8926	0.9345
	50/50	3	0.9152	0.9142	0.9161
	80/20	7	0.9125	0.8774	0.9475
	70/30	7	0.9289	0.9079	0.9498
	50/50	7	0.9107	0.8974	0.9240
	80/20	15	0.9215	0.8889	**0.9541**
	70/30	15	0.8975	0.8670	0.9279
	50/50	15	0.9008	0.8790	0.9227
	80/20	30	0.9163	0.8851	0.9475
	70/30	30	0.8942	0.8517	0.9367
	50/50	30	0.8973	0.8576	0.9371
GLM	80/20	3	0.8654	0.7931	0.9377
	70/30	3	0.8791	**0.8542**	0.9039
	50/50	3	**0.8815**	0.8469	0.9161
	80/20	7	0.8030	0.6552	0.9508
	70/30	7	0.8395	0.7161	0.9629
	50/50	7	0.7877	0.5819	**0.9934**
	80/20	15	0.8262	0.6820	0.9705
	70/30	15	0.8066	0.6240	0.9891
	50/50	15	0.7876	0.5896	0.9856
	80/20	30	0.7674	0.5479	0.9869
	70/30	30	0.8212	0.6598	0.9825
	50/50	30	0.8055	0.6202	0.9908
XGBoost	80/20	3	0.9180	**0.9310**	0.9049
	70/30	3	0.9037	0.8926	0.9148
	50/50	3	0.9150	0.9112	0.9187
	80/20	7	0.9190	0.9004	0.9377
	70/30	7	**0.9269**	0.9105	0.9432
	50/50	7	0.9112	0.9266	0.9189
	80/20	15	0.9193	0.9042	0.9344
	70/30	15	0.9088	0.8875	0.9301
	50/50	15	0.8971	0.8821	0.9122
	80/20	30	0.8462	0.7088	**0.9836**
	70/30	30	0.9066	0.8721	0.9410
	50/50	30	0.9001	0.8775	0.9227

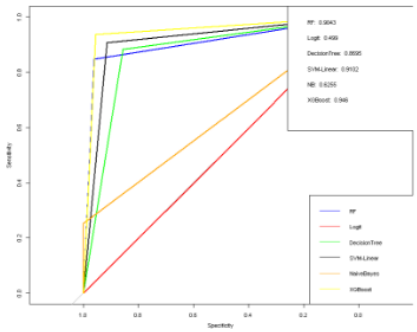
[Table 10: Test results with temporal dependence]



[Figure 10: ROC with temporal dependence and dimensionality reduction, 30-day interval]



[Figure 11: ROC with temporal dependence and dimensionality, 15-day interval]



[Figure 12: ROC with temporal dependence and dimensionality reduction, 7-day interval]

The models trained on the new feature set gave better results than the models trained on just last day's data. Amongst the six models trained the Random forest and XGboost models gave the most optimal accuracy and sensitivity. But considering the time needed to train the model, the bagging-based algorithm was faster to train and has scope to be parallelized as compared to the boosting method.

7. Future Enhancements

In this report we have limited our scope to only Seagate model number ST4000DM000, our analysis and prediction can be further extended to other hard drive models from Backblaze. Furthermore, we are now predicting the hard drive device failure on the day of its failure. If we could predict the device failure in advance, then suitable backup action can be taken to avoid the data loss. With the application of Deep Learning we can also come up with models which can self-analyse and predict the hard drive failure in advance.

8. Acknowledgement

We would like to thank ***Backblaze*** for open sourcing the data. Additionally, we would like to thank the core development team and contributors of the open source python package ***Tsfresh.***

9. Conclusion

In this report, we have analysed the Backblaze hard drive (Seagate - ST4000DM000) failure and used several prediction models for classification. We evaluated the prediction performance among Decision Tree, Support Vector Machine(SVM) – Linear, Naïve Bayes, Random Forest, Generalized Linear Model and Extreme Gradient Boosting (XGBoost) models across different intervals. We considered the temporal aspects and observed better results. The next most important task is to introduce these prediction models into the real world.

10. References

1. Hard Drive Data and Stats
2. Wikipedia S.M.A.R.T. metrics definition
3. False Discovery Rate
4. Optimizing rolling feature engineering for time series data

11. Appendix

Sample model training and evaluations are presented in this section. Interval = 30 days, split = 80/20

Cross Validation:

CART

2267 samples
2248 predictors
  2 classes: '0', '1'
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1814, 1814, 1814, 1813, 1813
Resampling results across tuning parameters:

cp	Accuracy	Kappa
0.03014354	**0.8773706**	**0.7556865**
0.07081340	0.8623800	0.7257696
0.65071770	0.7080054	0.3838804

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was cp = 0.03014354.

Support Vector Machines with Linear Kernel

2267 samples
2248 predictors
  2 classes: '0', '1'
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1813, 1814, 1813, 1814, 1814
Resampling results:

Accuracy	Kappa
0.8989799	**0.7957245**

Tuning parameter 'C' was held constant at a value of 1

Naive Bayes

2267 samples
2248 predictors
  2 classes: '0', '1'
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1813, 1813, 1814, 1814, 1814
Resampling results across tuning parameters:

usekernel	Accuracy	Kappa
FALSE	NaN	NaN
TRUE	0.6228443	0.1932852

Tuning parameter 'fL' was held constant at a value of 0
Tuning parameter 'adjust' was held
 constant at a value of 1
Accuracy was used to select the optimal model using the largest value.
The final values used for the model were fL = 0, usekernel = TRUE and adjust = 1.

Random Forest

2267 samples
2248 predictors
  2 classes: '0', '1'
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1813, 1814, 1814, 1813, 1814
Resampling results across tuning parameters:

mtry	Accuracy	Kappa
2	0.8372320	0.6646370
67	**0.9461787**	**0.8913123**
2248	0.9351528	0.8694538

Accuracy was used to select the optimal model using the largest value.
The final value used for the model was mtry = 67.

Generalized Linear Model

2267 samples
2248 predictors
  2 classes: '0', '1'
No pre-processing
Resampling: Cross-Validated (5 fold)
Summary of sample sizes: 1814, 1813, 1814, 1813, 1814
Resampling results:

Accuracy	Kappa
0.5342397	0.07688918

Test set:

CART

Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	233	14
1	72	247

Accuracy : 0.8481
95% CI : (0.8158, 0.8766)
No Information Rate : 0.5389
P-Value [Acc > NIR] : < 2.2e-16
Kappa : 0.6991
Mcnemar's Test P-Value : 7.923e-10
Sensitivity : 0.9464
Specificity : 0.7639
Pos Pred Value : 0.7743
Neg Pred Value : 0.9433
Prevalence : 0.4611
Detection Rate : 0.4364
Detection Prevalence : 0.5636
Balanced Accuracy : 0.8551
'Positive' Class : 1

SVM - Linear Kernel

Confusion Matrix and Statistics

Prediction	Reference 0	Reference 1
0	280	32
1	25	229

Accuracy : 0.8993
95% CI : (0.8715, 0.9228)
No Information Rate : 0.5389
P-Value [Acc > NIR] : <2e-16

Kappa : 0.797
Mcnemar's Test P-Value : 0.4268
Sensitivity : 0.8774
Specificity : 0.9180
Pos Pred Value : 0.9016
Neg Pred Value : 0.8974
Prevalence : 0.4611
Detection Rate : 0.4046
Detection Prevalence : 0.4488
Balanced Accuracy : 0.8977
'Positive' Class : 1

Naive Bayes
Confusion Matrix and Statistics
Reference
Prediction 0 1
0 305 209
1 0 52
Accuracy : 0.6307
95% CI : (0.5895, 0.6706)
No Information Rate : 0.5389
P-Value [Acc > NIR] : 6.093e-06
Kappa : 0.2114
Mcnemar's Test P-Value : < 2.2e-16
Sensitivity : 0.19923
Specificity : 1.00000
Pos Pred Value : 1.00000
Neg Pred Value : 0.59339
Prevalence : 0.46113
Detection Rate : 0.09187
Detection Prevalence : 0.09187
Balanced Accuracy : 0.59962
'Positive' Class : 1

Random Forest (t=51)
Confusion Matrix and Statistics
Reference
Prediction 0 1
0 293 17
1 12 244
Accuracy : 0.9488
95% CI : (0.9272, 0.9654)
No Information Rate : 0.5389
P-Value [Acc > NIR] : <2e-16
Kappa : 0.8968
Mcnemar's Test P-Value : 0.4576
Sensitivity : 0.9349
Specificity : 0.9607
Pos Pred Value : 0.9531
Neg Pred Value : 0.9452
Prevalence : 0.4611
Detection Rate : 0.4311
Detection Prevalence : 0.4523
Balanced Accuracy : 0.9478
'Positive' Class : 1

GLM
Confusion Matrix and Statistics
Reference
Prediction 0 1
0 0 11
1 305 250
Accuracy : 0.4417
95% CI : (0.4003, 0.4837)
No Information Rate : 0.5389
P-Value [Acc > NIR] : 1
Kappa : -0.039
Mcnemar's Test P-Value : <2e-16
Sensitivity : 0.9579
Specificity : 0.0000
Pos Pred Value : 0.4505
Neg Pred Value : 0.0000
Prevalence : 0.4611
Detection Rate : 0.4417
Detection Prevalence : 0.9806
Balanced Accuracy : 0.4789
'Positive' Class : 1