Artificial Intelligence Opportunities and an End-To-End Data-Driven Solution for Predicting Hardware Failures

by

Mario Orozco Gabriel

B.S. Chemical Engineering, Instituto Tecnologico y de Estudios Superiores de Monterrey, 2011

Submitted to the Department of Mechanical Engineering and MIT Sloan School of Management
in partial fulfillment of the requirements for the degrees of

Master of Science in Mechanical Engineering

and

Master of Business Administration

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2016

Author .....

Department of Mechanical Engineering and MIT Sloan School of Management

May 6, 2016

Certified by .....

Kalyan Veeramachaneni

Principal Research Scientist in Laboratory for Information and Decision Systems

Thesis Supervisor

Certified by .....

Tauhid Zaman

KDD Career Development Professor in Communications and Technology

Thesis Supervisor

Certified by .....

John J. Leonard

Samuel C. Collins Professor of Mechanical and Ocean Engineering

Thesis Supervisor

Accepted by .....

Maura Herson

Director of MBA Program MIT Sloan School of Management

Artificial Intelligence Opportunities and an End-To-End Data-Driven Solution for Predicting Hardware Failures

by

Mario Orozco Gabriel

Submitted to the Department of Mechanical Engineering and MIT Sloan School of Management
on May 6, 2016, in partial fulfillment of the
requirements for the degrees of
Master of Science in Mechanical Engineering
and
Master of Business Administration

Abstract

Dell's target to provide quality products based on reliability, security, and manageability, has driven Dell Inc. to become one of the largest PC suppliers. The recent developments in Artificial Intelligence (AI) combined with a competitive market situation have encouraged Dell to research new opportunities. AI research and breakthroughs have risen in the last years, bringing along revolutionary technologies and companies that are disrupting all businesses. Over 30 potential concepts for AI integration at Dell Inc. were identified and evaluated to select the ones with the highest potential. The top-most concept consisted of preventing in real time the failure of hardware. This concept was investigated using a data science process.

Currently, there exist a number of machine learning tools that automate the last stages of the proposed data science process to create predictive models. The utilized tools vary in functionality and evaluation standards, but also provide other services such as data and model storage and visualization options. The proposed solution utilizes the deep feature synthesis algorithm that automatically generates features from problem-specific data. These engineered features boosted predictive model accuracy by an average of 10% for the AUC and up to 250% in recall for test (out of sample) data.

The proposed solution estimates an impact exceeding $407M in the first five years for Dell Inc. and all of the involved suppliers. Conservatively, the direct impact on Dell Inc. is particular to batteries under warranty and is expected to surpass $2.7M during the first five years. The conclusions show a high potential for implementation.

Thesis Supervisor: Kalyan Veeramachaneni
Title: Principal Research Scientist in Laboratory for Information and Decision Systems

Thesis Supervisor: Tauhid Zaman
Title: KDD Career Development Professor in Communications and Technology

Thesis Supervisor: John J. Leonard
Title: Samuel C. Collins Professor of Mechanical and Ocean Engineering

Acknowledgments

This work was possible thanks to the help and contribution from many friends.

First, I would like to thank Dell for giving me the opportunity for this incredible internship in Austin. The team at Client Services was comprised of some of the most supportive people I have worked with. Many thanks to Nikhil Vichare, Leonard Lo, Steve Herington, Doug Reeder, Chad Skipper, Rick Schuckle, Jim White, Eugene Minh, Kevin Terwilliger, Catherine Dibble, Neil Hand, Charles Brooker, Christophe Daguet, and the rest of the CS team. Everyone's support went beyond their work obligations and played a key part in the development of this project.

A special thanks is in order to my supervisor Neal Kohl, who supported me since day one, helped me get familiarized with Dell and Austin, and made this experience a most positive one. He provided me with invaluable guidance, encouragement, and also demands to strive for excellence. Neal helped me define the scope, maintain focus, and set a strategy, among other crucial steps in the development of the project.

Thanks to my advisors, Kalyan and Tauhid, who guided me through a complex field. Their recommendations helped me navigate through this multifaceted area and grasp key points. Also, thanks to Kalyan, whose support was fundamental especially when developing this data science process and evaluating different technologies.

Thank you to MIT, LGO, and Mexico for allowing me to have enjoyed one of the most transformative experiences in my professional and personal life. I hope to provide someday such opportunities to more people and give back as much to the Institute, program, and my country. I would also like to thank all the LGO staff and LGO Class of 2016 who have made these two years unforgettable.

Finally, I thank my parents Mario and Julia for their instrumental advice and the values they have instilled in me through their way of life, and also thank my sisters Julie, Regina, and Natalia for always cheering me on. My family has always been my unconditional foundation of support and inspiration. Through their constant encouragement, I was able to maintain focus, sail through the difficult times, and find motivation to always continue doing my best. This work is dedicated to them.

Note on Proprietary Information

In order to protect proprietary Dell information, the data presented throughout this thesis has been altered and does not represent actual values used by Dell Inc. Any dollar values, hardware data, and product names have been disguised, altered, or converted to percentages in order to protect competitive information.

THIS PAGE INTENTIONALLY LEFT BLANK

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

List of Figures

THIS PAGE INTENTIONALLY LEFT BLANK

List of Tables

Glossary of Terms

AI	Artificial Intelligence
AGI	Artificial General Intelligence
ANI	Artificial Narrow Intelligence
AML	Amazon Machine Learning
API	Application Program Interfaces
ASI	Artificial Super Intelligence
AUC	Area under the curve
BP	Backpropagation
CPU	Central processing unit
CSV	Comma Separated Value
CT	Classification trees
DFS	Deep feature synthesis algorithm
DS	Data Science
DSM	Data Science Machine
FAR	False alarm rates, 100 times probability value
GB	Gigabytes
GBT	Gradient Boosted Trees
GLM	Generalized Linear Models
GPU	Graphics processing unit
HDFS	Hadoop Distributed File System
IoT	Internet of Things
IT	Information Technology
JSON	JavaScript Object Notation
KNN	K-nearest neighbor
NLI	Natural Language Interaction
NLP	Natural Language Processing
MARS	Multivariate Adaptive Regression Splines
ML	Machine Learning

MLaaS	Machine Learning as a Service
OCS	Operations and Client Services
PMML	Predictive Model Markup Language
RT	Regression trees
SGD	Stochastic Gradient Descent
SVM	Support Vector Machines
RAM	Random-access memory
RBF	Radial Basis Function networks
RDF	Random Decision Forests
RDS	Relational Database Service
REST	Representational State Transfer
ROC	Receiver operating characteristic
SAAS	Software as a Service
SAC	Support Assist Clients
SMART	Self-Monitoring and Reporting Technology
SPSS	Statistical Package for the Social Sciences
SQL	Structured Query Language
WA	Failure warning accuracy (probability)

Chapter 1

Introduction

With the purpose of exploring the exponentially growing field of Artificial Intelligence (AI) applications, this thesis has the objective to present an overview of AI technologies and machine learning tools available today, as well as their specific application for the prevention of hardware failures. AI is enabling the new "smart" hardware and software applications that are disrupting businesses and the way we live our daily lives. AI is a broad term that will be explained later, but for the objective of this work, AI applied to a pragmatic business-applied purpose will be referred to as "Smart Machines" [1]. This is highly relevant in our time, as barriers to entry have been considerably lowered by the availability of open source AI algorithms and platforms, the increase in economical and scalable cloud computing power, and surge in data generation, which can easily enable anyone to become a "citizen data scientist."

1.1 Motivation: if an oven can be smart, why can't a computer be smart?

Today, there is already a "smart" oven called June [2], which can precisely tell the difference between a chocolate and a raisin cookie, knows the weight, etc. to choose the right cooking time and temperature gradient to follow [3], while always improving its recipes. It has an NVIDIA central processing unit (CPU) and graphics processing

unit (GPU) along with two gigabytes (GB) of random access memory (RAM), Wi-Fi connectivity, a camera, and other sensors [2]. This is practically a computer. So, why aren't our computers even smarter?

The overarching goal behind this project is to explore the application of new data science tools that can be applied to different business cases and for Dell Inc.[1] to take advantage of such opportunities in an extremely demanding market. This research also explores potential concepts that will allow Dell to develop a competitive and higher-quality product. The proposed concepts create value for the customer and provide a better customer experience.

Currently, Dell has over 100 million deployed systems comprising desktops and laptops, and it is expected to ship around 20 million systems in the next year. Dell puts special emphasis to offer the most secure, manageable, and reliable products. Given this ambitious goal, there is always potential for improvement.

Today, computer failures are mostly treated post-event, which creates a major problem for the user and an important monetary impact on the service providers. However, Dell has recently launched Support Assist for Clients (SAC), software that enables the user to capture hardware data coming from hundreds of attributes pertaining to the hardware components, type of alerts, and failure types in a desktop or laptop.

The uses for this data can be quite varied and can provide valuable insights, such as system performance, influential interactions of hardware components and operation, etc. With this data, we can also potentially prevent hardware failures by warning the user, or even self-correcting based on real-time data that feeds into ever-improving predictive models. Just considering hardware cost and attending customers' calls, we have estimated these hardware failures to have an annual yearly cost of over $900M across the world. There is an interesting opportunity to make the products more reliable by making them predictive or proactive, rather than reactive, through an advanced and dependable process.

[1]Dell Inc. will be referred to as "Dell" throughout this work

1.2 Problem description and goals

Hardware component failures in desktops and laptops can occur instantly, randomly, and without much warning. This affects Dell's customers globally, and Dell itself. These failures can cause great inconvenience to the customer, going beyond the physical damage to the product and data by extending to invaluable losses in time, productivity, and critical activities that depended on the reliability of these products. Additionally, Dell allocates resources consisting of people, customer service organizations and facilities, and hardware components to replace the damaged parts. This entire infrastructure to remedy failures is also reflected in a heavy financial burden that has the potential of being reduced or, in the best scenario, eliminated, and transformed into a business opportunity.

This thesis, instead of conducting a root cause analysis for the hardware failures themselves, focuses on seeking a solution to make desktops and laptops effectively robust against hardware failures. This will be done through a thorough analysis of available machine learning tools and testing them with Dell's available data for hardware failures. There will be a special focus on the tools' functionalities, available machine learning algorithms, tuning of hyperparameters, evaluation metrics, and other services, such as visualization options.

1.3 Hypothesis

The hypothesis of this inquiry is that an appropriate model to predict hardware failures can be built using data collected from different hardware sensors in the computer. The expected outcomes are a variety of accurate models to be created with the different available machine learning tools. Additionally, if the first hypothesis is proven true, another hypothesis will be explored that states that labeled training data generated with the deep feature synthesis algorithm (DFS) [4] will result in higher accuracy models. Finally, after this analysis, the development of an end-to-end data-driven solution for the problem of preventing hardware failure will be proposed on hindsight.

1.4 Thesis overview

Chapter 1 introduces the motivation for this project. Some context is given for the problem to be solved: preventing hardware failures. Lastly, the hypotheses are stated as: determining whether hardware sensor data can be used to prevent different hardware failures, and that features created with the "deep feature synthesis" algorithm improve the accuracy of the predictive models.

Chapter 2 establishes the connection between data science, AI, and machine learning. It also describes the background and factors that are making AI an increasingly relevant and applied topic today. Dell's background and interest in improving products through AI is explained.

Chapter 3 explores different AI opportunities that are available for Dell. For this, a taxonomy and framework to understand AI is proposed and explained. In the last section, the opportunities within and outside of Dell are presented, and the importance of preventing hardware failure is discussed.

Chapter 4 reviews literature on previous work on the subject of preventing hardware failure. Statistical and machine learning methods are reviewed.

Chapter 5 proposes a data science process with six stages to follow an adequate path into solving prediction problems. The different stages and their connections are carefully explained. This section dives deeply into the available data, and especially the structure of the data, concerning hardware components in Dell's desktops and laptops. Also, the data science process is applied to Dell's particular problem to prevent hardware failures and focuses on hard drive failures due to their relevance and data availability.

Chapter 6 depicts seven different machine learning tools (platforms) and software that are currently available to analyze data, create predictive models, evaluate these models, and understand relationships and influences, among other uses. These tools are utilized to clarify their different capabilities, and specifically tested to build models to prevent hard drive failures using two different "labeled training data" approaches.

Chapter 7 highlights the financial impact that preventing hardware failure can

have across Dell and their suppliers. The last section emphasizes the value of the proposed end-to-end data-driven solution to prevent hardware failure in desktops and laptops on a global scale.

Chapter 8 concludes this thesis with a summary of the key findings and contributions and provides recommendations that resulted from this work. Future potential projects at Dell are discussed and conclusions are presented.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 2

Background

2.1 What are the connections between Data Science, Artificial Intelligence and Machine Learning?

Data science spans a broad range of interdisciplinary fields that includes computer science, mathematics, statistics, modeling, analytics, and information science, among others. The goal of data science is to utilize techniques to extract valuable knowledge from data with the aid of automated processes and systems. William Cleveland first introduced the discipline of data science in 2001, when he integrated the advances in "computing with data" to the field of statistics [5].

Like data science, AI is a broad subject and sometimes nebulous to people as it quickly convolutes into a topic dealing with machines that can think, reason, make decisions, and act like a human or have even higher capabilities than humans. Many researchers quickly turn to the psychology and philosophy of learning and discerning as humans and how this learning applies to these machines. More concretely, Tim Urban classifies AI into three categories [6]:

1. **Artificial Superintelligence (ASI)**

   ASI ranges from a computer that is somewhat smarter across the board than

a human to one that is smarter than any combination of all human society including scientific creativity, social skills, and general wisdom.

2. **Artificial General Intelligence (AGI)**
   AGI is also referred to as "Strong AI," or "human-level AI." It refers to a computer that is as smart across the board as a human–a machine that can perform any intellectual task that a human being can. This category of AI means machines have abilities to plan, reason, solve problems, think abstractly, comprehend the complex, and learn quickly and from experience (inference).

3. **Artificial Narrow Intelligence (ANI)**
   ANI is also referred to as "Weak AI." This AI specializes in just one area. This type already exists, and we use it in our everyday lives.

All of these capabilities make AI a wildly interesting topic; however, this thesis will focus on ANI technologies that are applicable to current businesses.

Machine learning is the most common technology, or set of algorithms, associated with AI as it has a very broad use. We clearly see the parallelism with AI as Andrew Ng, Chief Scientist at Baidu Research and professor at Stanford in the Computer Science Department, describes machine learning as "the science of getting computers to learn, without being explicitly programmed." There are two major machine learning branches: supervised and unsupervised.

1. **Supervised learning**
   Uses labeled data and focuses on classification and prediction. Some examples of algorithms include artificial neural networks, decision trees, genetic algorithms (evolutionary algorithms), K-nearest neighbor (KNN), multivariate adaptive regression splines (MARS), random forests, support vector machines (SVM), etc.

2. **Unsupervised learning**
   Uses unlabeled data and focuses on clustering, dimensionality reduction, and

density estimations. Some examples of algorithms include the Eclat algorithm, K-means algorithm, expectation-maximization algorithm, etc.

Each of these different branches has a plethora of different algorithms that vary in performance depending on the data and end goal.

In addition, deep learning has been an increasingly popular variant of machine learning. Deep learning uses neural nets typically with more than two processing layers. The layers provide a "deeper" level of features from the data, which provides better classification and prediction performance.

2.2 Three factors that are making AI very relevant now

John McCarthy coined the term "AI" in 1956 when the first academic conference on this subject was held [7]. AI went through an exploratory process with much of today's theory based on concepts from decades ago. So, why is it is so significant now? The crossroads of three factors have made AI stir unprecedented interest and activity in the past years and become very relevant in the business world. These factors are available data, increased computing power, and powerful algorithms.

Data generation has reached historically maximum levels, and most of the data generated is unstructured. As IBM states, "Every day, we create 2.5 quintillion bytes of data – so much that 90% of the data in the world today has been created in the last two years alone" [8]. Also, with the Internet of Things (IoT) on the rise, data is coming from an increasing variety of sources such as GPS, online posts, climate sensors, energy meters, transportation routes, the human body, etc. The amount of data and sources is expected to increase considerably. According to Strategy&, "the installed-base of internet-connected devices exceeded 7 billion in early 2014 and is expected to grow to 50 billion by 2020. More than 10 times the amount of personal computers" [9]. Even IDC mentions that by 2020, we will be producing 50 times more data than in 2011 as pictured in Figure 2-1 [10]. Therefore, we need systems that can

generate useful insight from all this data.

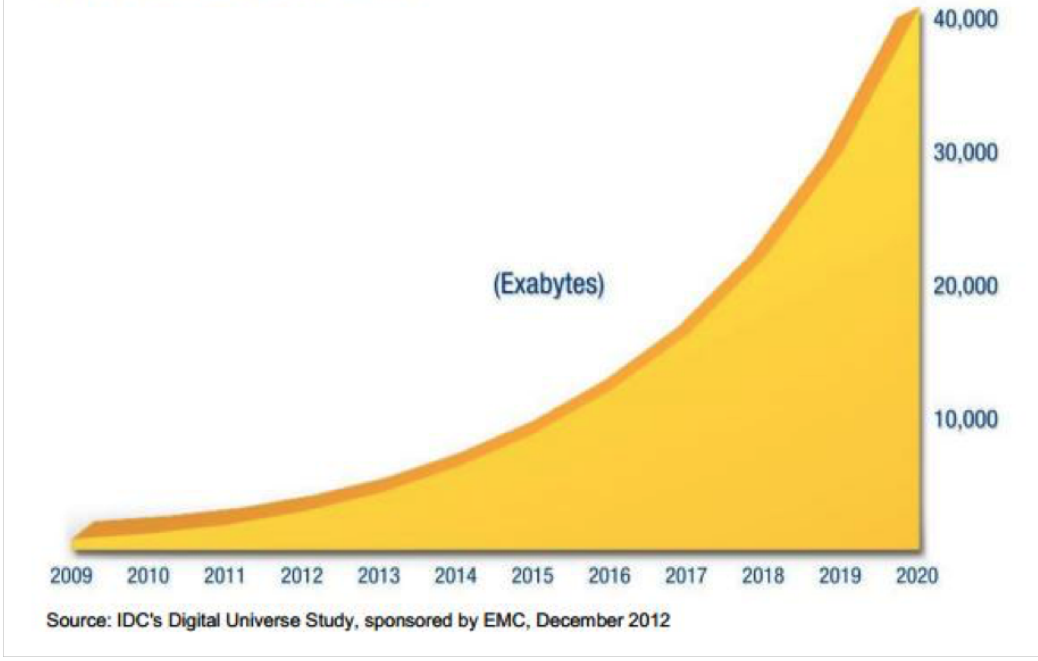


Figure 2-1: Exponential increase in data generation; 50-fold growth from 2010 to 2020. Greatly influenced by IoT.

Additionally, the computing power available for very low prices has played an important role in the processing of all the generated data. As seen in Figure 2-2 [11], the increase in computing technology follows an exponential growth curve, based on Moore's law, and futurist Raymond Kurzweil correlates this growth to a part of his "Law of Accelerating Returns" [12].

According to Deloitte, "computing cost-performance has decreased by more than three orders of magnitude in the past 25 years" [13], making it a decline of 33% year-on-year. Improvements in hardware, such as CPUs and GPUs, have allowed the efficient processing of this much data. According to Gartner [14], GPUs have had a 10,000 times improvement since 2008, increasing the number of possible connections from 1×10^7 million to 1×10^{11} million. An example is NVIDIA's new "Tesla GPU," which processes 10-100x the application throughput of traditional CPUs [15]. Moreover, advanced neuromorphic architectures based on field-programmable gate arrays surpass GPUs three times in energy efficiency and according to Gartner, have a 70% increase in throughput with comparable energy used [16].

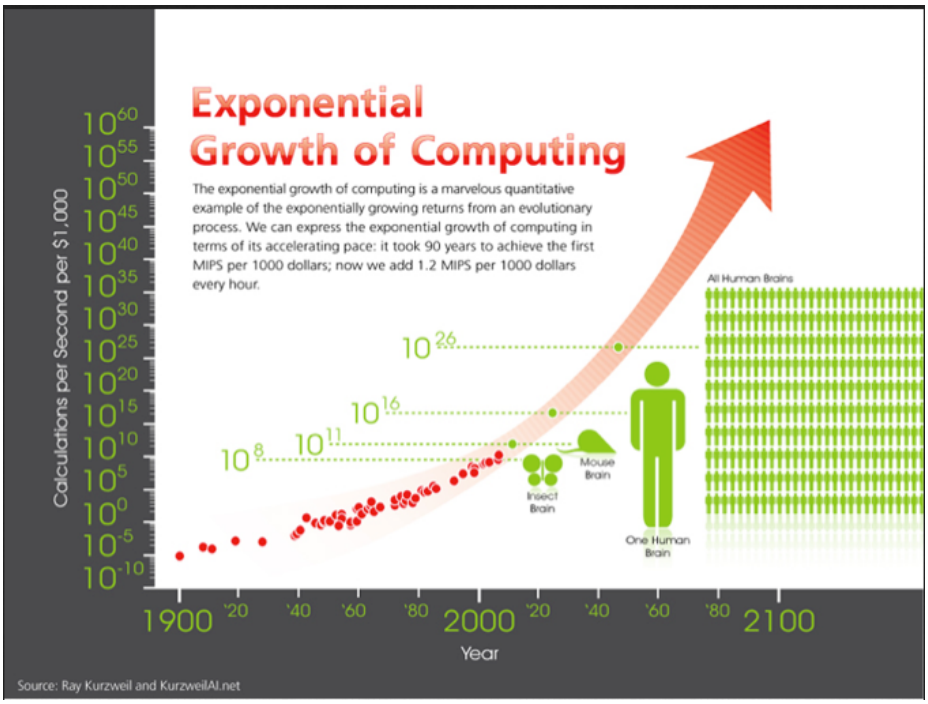


Figure 2-2: Increasing computing power as estimated by Ray Kurzweil. The *y-axis* describes the calculations per second (cps) per $1,000. As shown, $1,000 will be able to buy enough computing power as "one human brain" in the next few years.

Much of the structure of algorithms powering AI today existed years ago, but sufficient data and computer processing power were not easily available. Now, these algorithms have come to be used and proven very effective at tasks such as classification of structured and unstructured data, pattern detection, optimization, and predictive modeling, among other uses. Some examples of the types of algorithms used in AI are machine learning, deep learning, image recognition, natural language processing, etc. These algorithms have various uses such as data mining, text mining and search, expert systems, speech recognition and interaction, medical diagnosis, financial decisions, and fraud detection, among others. The key differentiators of these algorithms is that they are no longer programmed to "solve a problem," but to "learn how to solve a problem."

However, these algorithms have also improved, and we can see this strategy in the powerful open source algorithms that exist (such as in R or Python programming) and how many large companies have open sourced their algorithms, such as Google

(TensorFlow), Facebook (Torch), Apple (Swift), Microsoft (DMTK – Distributed Machine Learning Toolkit), Netflix (Spinnaker), etc.

With these enabling factors, automated procedures are being developed to gather and process data to derive valuable insights for actionable results. Given these factors, Gartner expects that "by 2017, the amount of citizen data scientists will grow five times faster than the number of highly skilled data scientists" [17].

The business community has taken notice. According to data retrieved from Gartner and Venture Scanner [18], over the course of 2015, more than 270 new startups focused on AI were founded and over $2 billion dollars were invested in the field, which is more than a 700% increase from the previous year, where it had even tripled as seen in Figure 2-3.

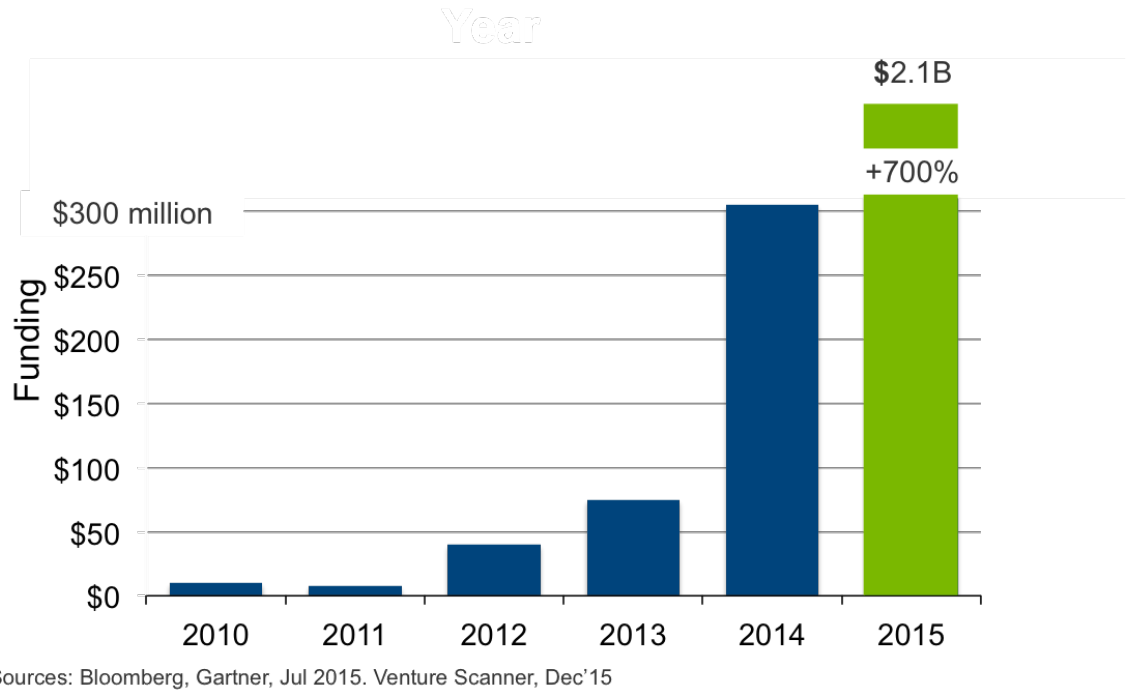


Figure 2-3: Total venture capital money for pure AI startups. The USA and London lead in start-ups, with many in Canada, India, and China. About one half of the funding has gone into deep learning, one fourth into computer vision, and one eight into natural language processing (NLP). The most active funds have been Intel Capital, Techstars, 500 Startups, and Y Combinator.

2.3 Dell Background

Looking further back, Dell's history [19] provides useful context for what are now applicable AI opportunities. Michael Dell founded Dell Computer Corporation, then PC's Limited, in 1984 with just $1,000 in his dorm room at the University of Texas at Austin. His vision was to change "how technology should be designed, manufactured, and sold." He set out to sell directly to customers with a focus on service in order to truly understand their needs. Just four years later, Dell went public and raised $30 million and continued growing 80% per year during the first 8 years, almost gaining half of the marketplace. During the 1990s, Dell expanded globally to Europe, Asia, and the Americas, while also becoming the number one ranked PC business in the USA and number one worldwide for PCs in medium and large businesses. In the 2000s, the ecommerce site dell.com became one of the highest volume ecommerce sites with over $40 million in revenue per day. Shipments grew 60%, about four times as much as typical for industry players. Later on, Dell started to focus on offering end-to-end solutions, which he achieved by acquiring over 20 companies for $13 billion from 2006 to 2013.

In 2013, Michael Dell bought back Dell. This event was at the time the biggest public company to return to private. Dell focused on four key areas: Operations and Client Services, Enterprise Solutions, Software, and Research. As always inspired to offer a great product to customers and with the ever-changing and fast-paced tech environment, Dell announced in 2015 plans to acquire EMC in the biggest acquisition in tech history for $67 billion [20]. Dell now has over 100,000 employees, and is one of the top three largest suppliers of PCs and laptops.

The focus of this work will be within the Operations and Client Services (OCS) business in Dell. OCS accounted for over half of the company's $60+ billion in revenue in 2013, and it consists of various main categories: notebooks, desktop PCs, thin clients, and client-related peripherals. OCS is the largest business unit within Dell, and with the tough market, Dell has been recently under constant pressure to innovate. Therefore, because of the nature of the market, any technologies or innovations that

will create a product with better performance and higher quality that can be priced above market will always be an attractive and also crucial opportunity to go after. This work focuses on the potential AI applications for desktops, notebooks, and thin clients.

Chapter 3

Smart Machines: AI Opportunities for Dell

3.1 What is a Smart Machine?

As explained in the introduction, this work focuses on applied AI for businesses. Gartner states, "'Artificial intelligence' and 'cognitive computing' are not synonymous with 'smart machines' and may set unrealistic expectations for development and deployment. 'Smart machines' reflect a more diverse, pragmatic business purpose" [1]. The "Smart Machines" term means that systems will acquire the ability to train themselves, learn from their mistakes, observe the environment, converse with people and other systems, enhance human cognitive capabilities, and replace workers in routine tasks. All these tasks are done through the different, applicable AI technologies this work has been referring to.

3.2 AI taxonomy

In order to better understand the opportunities that lie ahead, this work proposes a taxonomy to classify the different AI technologies and algorithms. The basis for this classification of technologies lies in their uses and applications. The proposed taxonomy for AI in this work is the following:

1. **Machine Learning**

   It can probably be considered the broadest and most applicable of the technologies due to its flexible interdependent nature. It is a subfield of computer science that derived from pattern recognition and computational learning theory. As previously explained, with this technology software can develop insights and features from data without it being explicitly programmed to do so. As mentioned, the two main branches are supervised and unsupervised learning (see 2.1).

2. **Deep Learning**

   It is a set of algorithms that coincide with machine learning. More specifically, it is a technology based on neural nets. The concept of neural nets was inspired by the biological functioning of the brain, which takes multiple inputs and different areas extract specialized parts of information. Deep learning models abstract data from more complex sources and utilize multiple "specialized layers," which permit a deeper level of feature abstraction, such as prediction accuracy and classification.

3. **Image Recognition**

   It is also known as computer vision. It is the field that has specialized methods to gather, process, analyze, and understand images through particular algorithms. It utilizes a combination of physics, geometry, statistics, and learning theories in order to achieve recognizing images.

4. **Natural Language Processing and Natural Language Interaction (NLP & NLI)**

   It is the ability of a computer to understand text and human speech. This can be done through a combination of machine learning algorithms in addition to specific rules to follow. This allows a computer to understand the structure, extract meaning, process it, and produce natural language.

5. **Prescriptive Analytics**

It is the highest of the three levels of analytics. It combines advanced statistics and mathematics with data synthesis and computational science. *Prescriptive analytic*s will make different predictions based on the data and then recommend different choices of action to implement depending on the expected outcomes and consequences. Second, *predictive analytics* involve advanced forecasting methods such as regression, which predict what will happen according to the data we have. Third, *descriptive analytics* is a term used for performing the basic analytics to understand past events and can generate information such as average, mean, median, standard deviation, etc.

3.3 AI framework

In order to detect potential opportunities for Dell, a framework was developed to understand and classify AI as well as to understand market potential. The methodology to develop such framework was based on interviews and research. The proposed AI framework is layered in the following way:

1. **Smart Infrastructure**

   Includes the client (personal computers and laptops), storage, servers, networking, database and cloud management. This provides advanced infrastructure that scales beyond current capabilities of IT, where it is agile, simple, and automated for dynamic environments.

2. **Smart Data**

   Includes discovery and analytics software, cognitive platforms, and information management software. This layer is based on the platforms and tools such as machine learning, deep learning, prediction, visualization, etc. and combines features of databases, business intelligence, and comprehensive research.

3. **Smart Apps and Services**

   Includes end-user apps and services. This layer provides real-time insights

and recommendations, automation and enhancement of work, and predictive behavior.

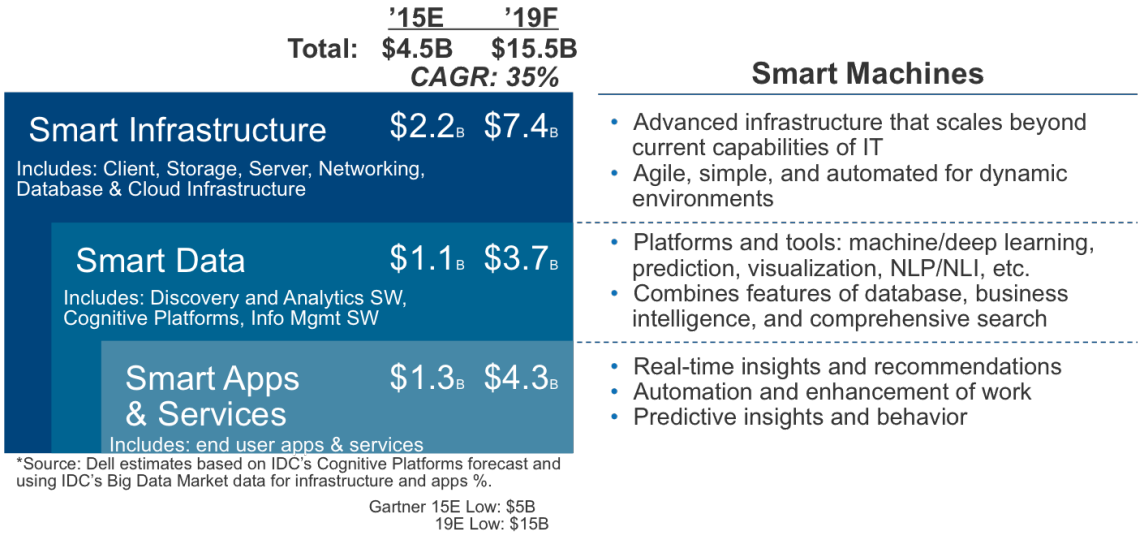


Figure 3-1: AI Framework and Market Potential. Today's estimated $4.5B market is growing at 35% CAGR based on IDC's Cognitive software platform forecast.

Within the proposed framework, Dell clearly plays in the first and second layers of Infrastructure and Data, which importantly includes a large part of the market. However, it is important to capitalize on the "Smart" opportunities within these fields as shown in Figure 3-1.

3.4 Opportunity detection

Having defined a taxonomy and framework for AI, two strategies, internal and external, were set to research opportunities for Smart Machines. The internal strategy was to gather all current Smart Machines or related AI initiatives as well as to conduct a brainstorming session with technical experts. The external strategy was to conduct an extensive market and competitive analysis to detect attractive use cases.

3.4.1 Opportunities within Dell

A brainstorming session was conducted and later evaluation criteria were selected and thoroughly researched in order to prioritize the implementation of the concepts. The concepts were focused on the Operations and Client Services business at Dell.

The brainstorming yielded 30+ potential concepts, which were classified in the following categories: security, serviceability, manageability, and productivity. The criteria used to evaluate these concepts consisted in: technology readiness including ease of implementation, financial impact, Intellectual Property (IP) potential, and business alignment within Operations and Client Services and Dell. For more details on the ranking, reference Appendix A. Further analysis revealed that 16 were completely new concepts, seven were in discussion processes, and the rest were already being developed within Dell in some form.

3.4.2 Why monitor hardware to prevent failures?

After evaluating the different concepts, only brand new ideas were selected, as shown in Figure 3-2, and out of those, the top ranked concept from each category was selected for further investigation. These were:

1. **Security** – Environmental and contextual security
2. **Serviceability** – User self-help smart Q&A
3. **Manageability** – Self-management and self-healing (hardware failure prevention)
4. **Productivity** – Personal productivity enhancer

Monitoring hardware for failure prevention stood out among the rest for several reasons consisting of: technology readiness (including data availability), financial opportunity, and Dell fit. On the technology side, Dell had started to actively collect sensor data from authorized systems, which included desktops and laptops, within the past year. This data consisted of the parameters within hardware sensors, which

		Parameters					
		Tech-readiness (feasibility / ease of implementation) 1 - 4 5 yrs - ready	Return / Impact * Financial 1 - 4 low - high (20+%)	Return / Impact * Reputation 1 - 4 low - high	Return / Impact * IP 1 - 4 low - high	CS/Dell Fit (Business Alignment) 1 - 4 low - high	Total 1 - 4 low - high
Security	Environment & Contextual Security	CONFIDENTIAL					
Serviceability	User self-help Q&A						
Manageability	Self-management & healing						
Productivity	Personal Productivity Enhancer						

Figure 3-2: The categories and top-concept applications were evaluated and selected using the shown parameters.

are collected by the Support Assist for Clients software. Algorithms, such as machine learning to find failure patterns, were also readily available. Financially, this concept promised a very attractive opportunity, and seemed to be viable to implement in the short term. Lastly, there was alignment with two of Dell's quality drivers as most manageable and most reliable to give the user a supreme customer experience.

3.4.3 Opportunities outside of Dell

The global market for Smart Machines was also considered when evaluating opportunities and, for this, a thorough market and competitive analysis was performed. Using the established AI taxonomy, research was done to evaluate rising and also working concepts from leading companies and startups.

To evaluate the potential for opportunities, companies, as shown in Figure 3-3, were evaluated against four criteria:

1. **Product availability**

   Low consisted of only trials and demos. Medium consisted of already working with less than 10 clients. High consisted of 1000+ users and/or 10+ use cases.

2. **Company or product maturity**

   Low equals that the company has been established between zero and three years. Medium refers to a timeframe between four and seven years. High is between eight and 14 years, and very high is 15+ years.

3. **Size of company or product**

   Very low is established as less than $5M in revenue or less than $10M in funding.

Low is between $5-$50M in revenue or between $10-$20M in funding. Medium is between $50-$500M in revenue or between $20-$100M in funding. High is between $500M-$5B in revenue or between $100-$500M in funding. Very high consists of $5B+ in revenue or $500M+ in funding. Growth rate of revenues was also considered.

4. **IP**

   Very low is established as less than 5 patents. Low is between 6-10 patents. Medium is between 11-50 patents. High is between 51-200 patents. Very high consists 200+ patents.

Figure 3-3: Summarized AI Competitive Analysis. There was a great deal of activity in "cognitive" platforms, "data" machine learning, and analytics. Some of the most relevant companies are shown in the figure.

The conclusions on this competitive analysis show that the majority of the compa-

nies or products are very new and have been in the market for less than three years. Additionally, platforms are lowering the barrier of entry to the market as algorithms and technologies for Smart Machines are available on-demand to the public. Moreover, startups are focusing on a specific vertical to perfect their technology.

The results demonstrated clear opportunities exist for Dell within "Data" machine learning as already-developed products exist and the maturity of the companies means there is room to grow and become an important player in this area.

Chapter 4

Literature Review

In this chapter, seven relevant research papers are reviewed. These research papers deal with prediction models and methods that have been used for failure prediction in hardware systems, hard drives, and other computer systems. The prediction methods described in the papers involve statistical analysis and a variety of machine learning algorithms. A brief analysis of the work and conclusions of each will be presented in order to better understand the progress made in this subject.

4.1 Statistical methods in hardware failure

Elkan and Hamerly [21] utilize naïve Bayesian Classifiers in order to overcome the difficulty of lack of data, since hard drives fail approximately 1% per year. The naïve Bayes method is a recognized supervised learning method that produces a classifier able to distinguish between two classes of data. Elkan and Hamerly studied the Self-Monitoring and Reporting Technology (SMART) system in hard drives, which is a failure prediction system to predict near-term failure. Typical SMART data includes variables such as power-on hours (POH), contact start-loops (CSS), seek errors in track servo (SKE), spinup time (SUT), etc. The SMART system can be regarded as a statistical hypothesis test based on each individual manufacturers' developed thresholds. These thresholds are set based on testing and engineering knowledge about the operational parameters. According to Hughes et al. manufacturers estimate the

"failure warning accuracy" (WA), or true-positive rate, of these systems to be between 3-10%, with estimated 0.1% "false alarm rates" (FAR), or false-positive results [22]. The failure events in hard drives do not happen very often, which means a known statistical distribution is hard to achieve. Elkan and Hamerly achieved a 33-59% WA with 0.5-0.7% FAR, which is a higher WA than the typical SMART performance, but with higher FARs.

Hughes, Kreutz-Delgado, and Murray also analyze [22] the SMART system in hard drives and propose a method that uses a distribution-free Wilcox rank-sum statistical test, since the problem deals with a rare-occurring event. This statistical test is recommended when failures are rare and false-positives are very costly. The proposed rank-sum method is used in combination with multivariate and ORing tests. The multivariate tests exploit the statistical correlations between attributes, and the ORing test simply uses a single attribute. The researchers achieve a 40-60% WA with 0.2-0.5% FAR, for both the multivariate and ORing tests [22].

4.2 Statistical and machine learning methods in hardware failure

Going forward, Murray, Hughes, and Kreutz-Delgado compare statistical and machine learning methods [23] to try to predict hard-drive failures in computers by utilizing special attributes from the SMART system. They compare non-parametric statistical tests (rank-sum, reverse arrangements, and a proposed algorithm) as well as machine learning methods (SVMs and unsupervised clustering). They propose a new algorithm based on the multiple-instance learning framework since this study is considered a two-class semi-supervised problem. Multi-instance learning deals with objects that generate many instances of data, and an object's data is collected in a "bag," which receives a discrete value of 0 or 1 depending on the prediction problem. Additionally, they use a simple Bayesian classifier and pair it with the multiple-instance framework to create the multiple instance-naive Bayes (mi-NB) algorithm. Feature selection for

the models is done through the statistical reverse arrangements test and selecting such features depending on their relevant z-scores. The prediction models are created with SVMs, unsupervised clustering (Cheeseman and Stutz' Autoclass package [24]), rank-sum tests, and the mi-NB algorithm. They ran experiments using 25 attributes, single attributes, and a combination of attributes. The results show that SVMs provide the best performance with 51-70% WA and 0-6% FAR, followed by the rank-sum tests with 28-35% WA and 0-1.5% FAR, mi-NB with 35-65% WA with 1-8% FAR, and clustering with 10-29% WA with 4.5-14% FAR. Murray et al. highlight that the achieved results of non-parametric statistical tests are to be noted, since they come with great computational efficiency considering that SVMs take 100 times longer in training and testing [23].

Motivated by the growing complexity and dynamism of computer systems, Salfner, Lenk, and Malek [25] report a survey of over 50 methods for failure prevention in hardware. They developed a taxonomy for failure prediction approaches for hardware and software, which can be represented with the following stages:

1. **Failure Tracking**

   It is based on the occurrence of previous failures. Methods included are probability distributions and co-occurrence.

2. **Symptom Monitoring**

   It is based on periodical analysis of the system. The key concept is that through system monitoring, the side-effects of individual hardware degradation, which are very difficult to catch, can be detected. Methods included are function approximations (stochastic models, regression, and machine learning), classifiers (Bayesian and fuzzy), system models (clustered models and graph models), and time series analysis (regression, feature analysis, and time series prediction).

3. **Detected Error Reporting**

   It is a non-active method, driven by events logged in the input data when an error occurs. Methods included are rule-based approaches, pattern recognition, statistical tests, and classifiers.

4. **Undetected Error Auditing** is actively searching for incorrect states of the system that can potentially predict failures within the system.

Each method applies differently depending on the prediction problem and use case the user wants. Salfner et al. [25] do not provide remarks on WA or FAR of the different methods as the main purpose is to provide a framework for methods and their use cases. They conclude that proactive fault management will be the key that enables the next generation of dependability improvements.

4.3 Machine learning methods in hardware failure

Turnbull and Alldrin [26] explore the prediction of hardware failure for servers with the hypothesis of being able to use sensor data for such predictions. To predict failures they use the servers' sensor logs for positive (failed) and negative (did not fail) cases. Each log is composed of individual entries that are recorded every approximately 75 seconds and contain sensor data, such as different board temperatures and voltages. These entries also record any failures. The feature vectors from these entries are extracted during a "sensor window," which is a specific amount of time when entries are collected. Each feature vector is associated with a "potential failure window" that comes after the "sensor window" as time continues. These vectors are used as input in a Radial Basis Function (RBF) network, a type of a neural network, to solve a two-class classification problem. They use 18 hardware sensors to build 72 features that include the mean, standard deviation, range, and slope. They achieve an 87% WA with a 0.1% FAR. They conclude that sensor data can be used to predict failures in hardware systems. An important remark they also make is to always consider the cost of prediction accuracy in false positive rates as well as working in a context of infrequent hardware failure [26].

Zhu et al. [27] provide a new approach to predict failure in large scale storage systems through SMART data analysis. They propose a new Backpropagation (BP) neural network and an improved SVM that achieve higher WA with considerably low FARs. They utilize 10, out of the 23, SMART attributes and their change rates as

features. The SVM is built using LIBSVM [28] and achieves a 68.5-95% WA with a 0.03-3.5% FAR. The BP neural network is composed of three layers and 19, 30, and 1 nodes, respectively, with a target value of 0.9 for healthy drives and 0.1 for failed drives in the output layer node. This BP neural network model achieves a 95-100% WA with 0.5-2.3% FAR. Their proposed models achieve very high WA with some tradeoffs in the FARs.

One of the most recent works in hard drive failure prediction developed by Li et al. [29] involves Classification Trees (CT) and Regression Trees (RT). They achieve some of the best WAs with low FARs, and provide additional capabilities for the hardware's health assessment. To build the trees for the models, they utilize the SMART attributes, as well as their change rates, and their targets' "good" or "failed" values as input features. For their models, they utilize "information gain" as the splitting function for the nodes. This split function looks through all the SMART attribute values and finds the best split that maximizes this "gain in information." The gain in information is calculated by comparing the entropy of information that each node contains. They continue to grow the tree by recursive partitioning until a node contains only one class or the node will not satisfy the split conditions. Additionally, the RT model is built similarly to the CT, but instead of using the "information gain" to split, they utilize "minimum squares" about the mean. Also, the RT model uses a quantitative target value rather than a classification of "good" or "failed" to describe the drive's health status. The CT model achieves a 94-96% WA with .01-.1% FAR and the RT model achieves a 63-96% WA with .01-.15% FAR.

As seen in the previous section, the progress for predicting hardware failure encompasses many areas and great advances and work have been done particularly in modeling hard drive failures. Different statistical and machine learning methods are used depending on the data availability, computing processing efficiency, and the type of prediction problem to solve.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 5

The Data Science Pipeline

Having developed a solution and undertaken a systematic approach, we use this experience to explicate a data science process, in order to understand the different stages and steps required to go from a collection of data to a prediction model. Overall, the proposed process consists of six stages and steps, as shown in Figure 5-1. In the following subsections, we will go through each stage and explain it in the context of our particular goal: building a predictive model for hardware failure.

5.1 Raw data

In a typical enterprise, data is generated by a diverse array of actors including sensors, people, and computers, and appears in a variety of different formats, ranging from logs to databases to flat files. This data is usually stored with no particular goal in mind. It is often either unorganized, or organized in such a specific manner that it cannot be broadly generalized.

In our particular scenario, Dell generates raw data from many sources. We will focus on one in particular: customers who have authorized the collection of sensor data. This data is generated from hundreds of hardware sensors inside computer systems and is then uploaded to a database. One of the main databases containing this type of raw data is called DDSAC, or "Dell Data – Support Assist Client." Table 5.1 lists a summary of the characteristics from a snapshot of the DDSAC database between

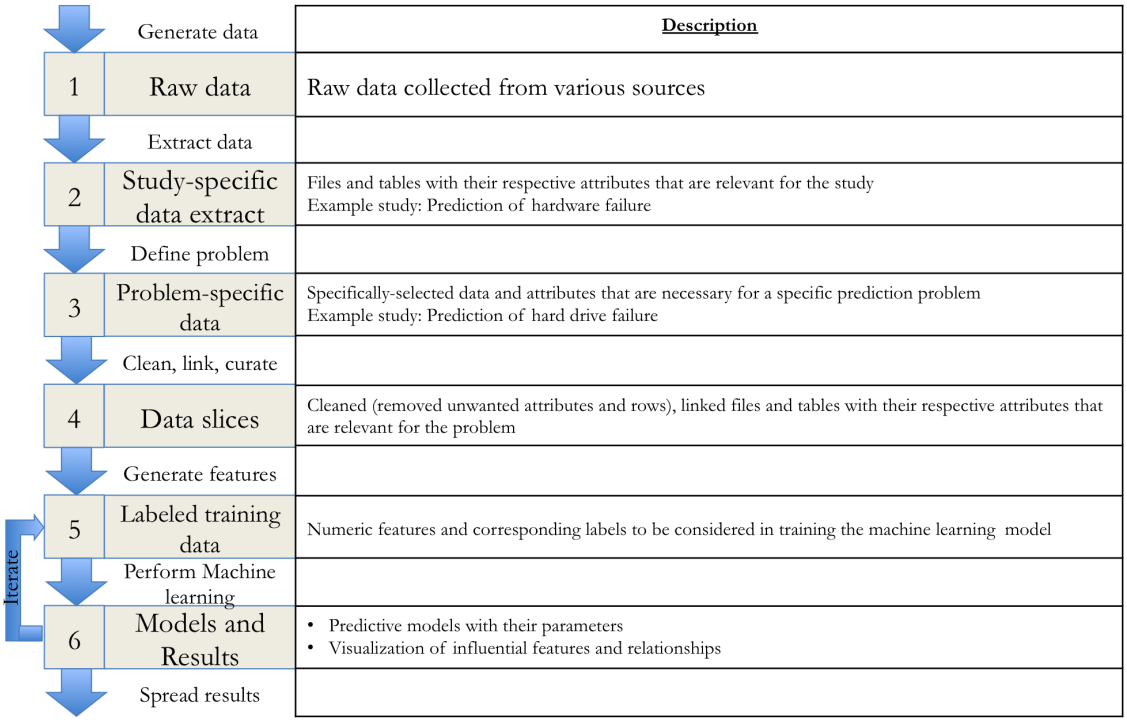


Figure 5-1: Proposed data science process.

February 2015 and November 2015.

Tables	40
Total Entries (Rows)	23,588,020
Data Space	3,571 MB

Table 5.1: DDSAC Summary between February 2015 and November 2015.

This data is organized in many different tables within the DDSAC database. Table 5.2 lists a few additional databases capturing data in Dell's SQL servers.

5.2 Study-specific data extract

Depending on which problem is of interest, this raw data could have many different uses. For example, one data scientist might be interested in *predicting* hardware failures, while another might wish to *predict* usage patterns. Solutions to both of

	Stage	Files	Size (rows)	Size (MB)
1	Raw data	DCSAI	153.3M	10,344
		DDSAC	23.6M	3,571
		Dispatch_Information	23,904	1
		SMART	1.5M	617

Table 5.2: A set of databases in Dell's SQL servers. DCSAI (Dell Client Support Assist Internal), DDSAC (Dell Data Support Assist Client), Dispatch_Information (dispatched hardware components for DDSAC), SMART (Hard drive sensor readings).

these problems could be derived from the same raw data. Once a particular goal is chosen, a data scientist's first task is to extract data that is applicable to the relevant study or question.

With a specific goal in mind–*predicting hardware failures*–we selected specific data tables that could contain useful information for solving this problem. In this particular case, we chose DDSAC[1], which contains the authorized external customers' data. For these customers, we also have the Dispatch_Information table, which has specific data on hardware dispatches, or shipments, for these customers. Since these dispatches are usually called in to replace a failed or malfunctioning component, we considered this information to imply that the component failed prior to its dispatch date.

We then investigated the DDSAC database to select a subset of the 40 available tables. Below, we describe some key points of the data, in order to to better understand its structure and characteristics:

1. **ID that uniquely identifies a computer system**

   Computer_ID is an attribute that uniquely identifies a specific system (a desktop or laptop) and its subcomponents across time.[2]

2. **Tables with sensor data associated with each hardware component**

   Within DDSAC, there is a corresponding table with sensor data specifically for

[1]In contrast, we could not use an even bigger database–DCSAI, which holds seven times the amount of data points than DDSAC (over 150M rows vs. 23M rows). This data in DCSAI is from Dell's internal employees computer systems. Even if it shows Alerts, there was no dispatch information available for these computer systems or no way to detect an actual failure. We selected DDSAC due to this associated dispatch information, which we considered to be a proxy for failure.

[2]This information does not identify the user but only the specific hardware. It is classified as non-personally identifiable information (non-PII) at Dell and in the industry widely.

each hardware component being monitored. For example, Disk.dbo is a table for the hard drive that contains data from different sensors monitoring its use.[3] Similarly, each table within the database contains a variety of fields; combined, they contain about 450 different fields. These tables either have a Computer_ID or a DataPoint_ID, which allows us to link them to a specific system.

3. **DataPoint_ID**

   A DataPoint_ID uniquely identifies a collection of data[4] at a specific time point. The DataPoint_ID is a key identifier, and exists in approximately 95% of the tables. In many cases, DataPoint_ID is recorded alongside Computer_ID, thus identifying exactly which system the data was collected from. When Computer_ID is missing in any of the tables, it can be retrieved by finding all the occurrences of the corresponding DataPoint_IDs and finding an associated Computer_ID in any of these occurrences. Thus, we are always able to identify which computer system a particular data point belongs to.

4. **SensorData table**

   Even though there are individual tables recording the sensor data corresponding to the usage of an individual hardware component, the SensorData table has the data from different components' sensors. The SensorData table has 132 different fields, which contain readings from sensors as well as basic sensor and system information. One of these fields is DataPoint_ID.

   In the SensorData table, there are 919,667 DataPoint_ID records (258,850 records are from desktops, 660,744 are from laptops, and the rest are unclassified). There are 342,287 unique Computer_IDs. 101,649 of these are desktops, 240,607 are laptops, and the remainder of them are unclassified.

5. **Dispatch table**

   The other essential table for predicting hardware failure is "Dispatch_Information."

[3] ePPIDs (Electronic ID) allow us to uniquely identify a subset of these individual components. For example, hard drives, batteries, wireless cards, and motherboards are tracked, while cables, fans, and monitors are not.

[4] A collection of data is sensor readings at or up until that time point.

This is a list of parts dispatched to a customer, and includes the Computer_ID, the dispatch date, and the dispatched parts. Each Computer_ID may have one or multiple dispatched parts (e.g. motherboard and hard drive).

The Dispatch_Information table contains 7,348 unique[5] Computer_IDs. These 23,094 dispatch events come from the 342,287 total tracked computers on DDSAC, which shows around a 2.15% failure rate in about 9 months time. As can be seen in Figure 5-2, hard drives and motherboards are two of the parts most affected by hardware failure.

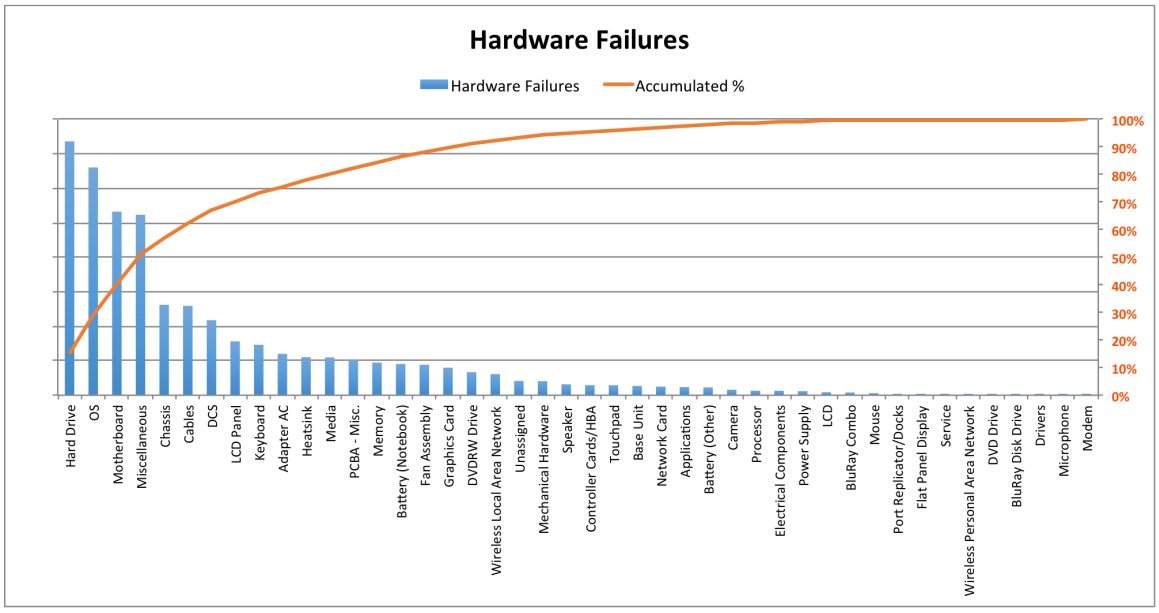


Figure 5-2: The frequency of dispatches of different components closely follows the *Pareto principle*. As seen, hard drives are the most frequently dispatched component.

Data collection process: As shown in Figure 5-3, hardware sensor data is generated and recorded through two types of events: continuous and discrete. In a continuous process, data is generated and recorded in the corresponding tables on a daily basis. In the discrete process, data is generated only when there is a trigger, such as an alert or failure, and is recorded in corresponding tables, such as the SensorData or Alert tables. Currently, all this recorded data is uploaded to the SQL server only

[5]Out of the 23,094 total dispatch events, which do not necessarily mean failures, but we will consider hard drive dispatches as a proxy for hard drive failures.

when there is a trigger, leaving much of the data on the systems. Thus, we only have DDSAC data when there have been triggers and alerts.

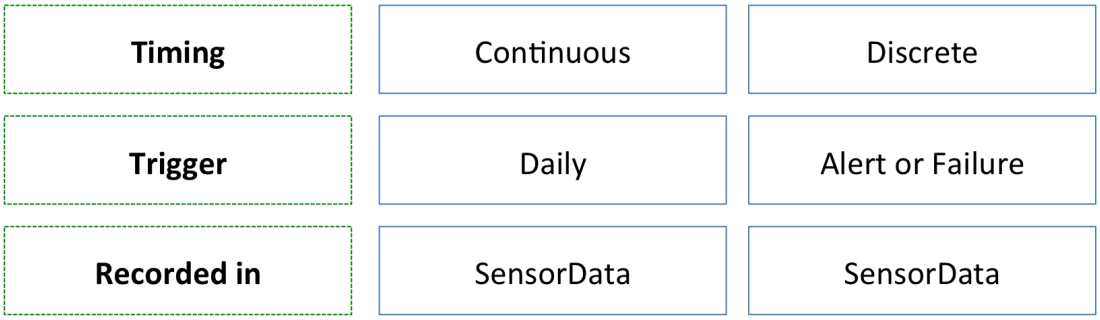


Figure 5-3: There exist practically two data generation and acquisition paths that are triggered by a daily rule or an unexpected alert or failure.

To recap:

- Our goal is to predict hardware failures.
- This study considered all data captured in SQL servers from February 2015 through November 2015.
- We selected the DDSAC database because dispatch information was available for the computers in this database.
- Of the 40 different tables contained within the DDSAC database, for building predictive models for hardware failures, we selected just 25. Table 5.3 shows the characteristics of these tables.

	Stage	Files	Size (rows)	Size (MB)
2	Study-specific data extract	• dbo_Alert.csv • dbo_AlertDetail.csv • dbo_BTModule.csv • dbo_Battery.csv • dbo_Bios_Internal_Logs.csv • dbo_Cable.csv • dbo_CableChangeHistory.csv • dbo_CrashInfo.csv • dbo_DIMM.csv • dbo_Disk.csv • dbo_DTFan.csv • dbo_HardwareChangeLog.csv • dbo_LanAdapt.csv • dbo_LogicalProcessor.csv • dbo_Monitor.csv • dbo_MSBugCodes.csv • dbo_NBFan.csv • dbo_Partition.csv • dbo_SensorData.csv • dbo_SMART.csv • dbo_SystemUnit_QITeam.csv • dbo_SystemUnit.csv • dbo_Thermistor.csv • dbo_WlanAdapt.csv • dbo_WWAN	• 356,858 • 111,392 • 311,307 • 89,758 • 452,049 • 2,346,483 • 16,475 • 146,819 • 1,950,698 • 1,333,304 • 171,976 • 732,211 • 917,052 • 907,378 • 4,271,891 • 899,053 • 265 • 320,623 • 4,512,656 • 1,125,679 • 225,635 • 341,285 • 1,232,045 • 816,499 • 557	• 121 • 21 • 112 • 14 • 64 • 226 • 2 • 56 • 254 • 203 • 16 • 79 • 1,024 • 133 • 134 • 86 • 0 • 27 • 293 • 415 • 13 • 29 • 125 • 126 • 0

Table 5.3: Study-specific data extract description.

5.3 Problem-specific data

Our next stage is to define a specific predictive problem. We have a number of options: for example, we could predict hard drive, motherboard, or battery failures. We chose to predict hard drive failures because this task fits three criteria:

1. The frequencies of dispatch for different components, as shown in Figure 5-2, show that hard drives are the most commonly dispatched component.

2. The hard drive is a critical system component containing locally-stored data and programs.

3. It provides the maximum possible *training examples* available for training a model.

For this specific prediction problem, two tables, Dispatch_Information and Sensor-Data, are used. The Dispatch table provides data on hardware components shipped due to failure. To reiterate:

1. The Dispatch table gives us the time point when a hard drive was dispatched to a customer.

2. The SensorData table provides information regarding how the specific system (identified via Computer_ID) was operating (through multiple data collection points identified by DataPoint_IDs) before it failed.

Table 5.4 shows a description of the files finally used for building predictive models for hard drive failures. Table 5-4 and Table 5-5 show a snapshot of these two files.

	Stage	**Files**	**Size (rows)**	**Size (MB)**
3	Problem-specific data	• newpd_log_data_large.csv • DispatchInfo_Parsed.xls	• 727,740 • 23,904	• 633 • 1

Table 5.4: Problem-specific data description.

		Date					
ComputerID	ComponentID	Begining	Ending	Sensor 1	Sensor 2	·····	Sensor 114
X432121 X812114 ZK1434 . . .							

Figure 5-4: Problem-specific data file (newpd_log_data_large.csv). It has the Computer_ID, DataPoint_ID, timestamps, and readings from multiple sensors.

ComputerID	Dispatch date	Commodity
X812114 ZZ4614 . . .	12/14/14 11/1/15 . . .	Hard Drive Motherboard . . .

Figure 5-5: Problem-specific data file (DispatchInfo_Parsed.xls). It has the Computer_ID, Dispatched Commodity (e.g. hard drive, motherboard, etc.), and the date of dispatch.

5.4 Data slices

Cleaning and curation: Next, we prepare the data set for feature engineering and machine learning. First, the files need to be cleaned, linked, and curated. This process consists of the following steps:

- Removing columns that had no data.
- Removing columns that had no relation to hard drive failure.
- Standardizing date and time formats.
- Eliminating empty rows.
- Linking, or reconciling files that had the same attributes under different names (such as Computer_ID and ComputerID).

Slicing: After curation and linking, we slice the data to remove unusable data or data that could interfere with the prediction model. Specifically, for each Computer_ID we remove the sensor readings that correspond to time after the dispatch date. Thus for each Computer_ID, we construct a data slice that has all the sensor readings prior to the dispatch date. Figure 5-6 shows the process of slicing and removing the post prediction data. Table 5.5 shows a summary of the sliced files, which are then taken through to the next stages.

	Stage	**Files**	**Size (rows)**	**Size (MB)**
4	Data slice	• Sensor_data_aggregated.csv • Outcomes.csv	• 42,313 • 12,483	• 4 • 0

Table 5.5: Data slice file description.

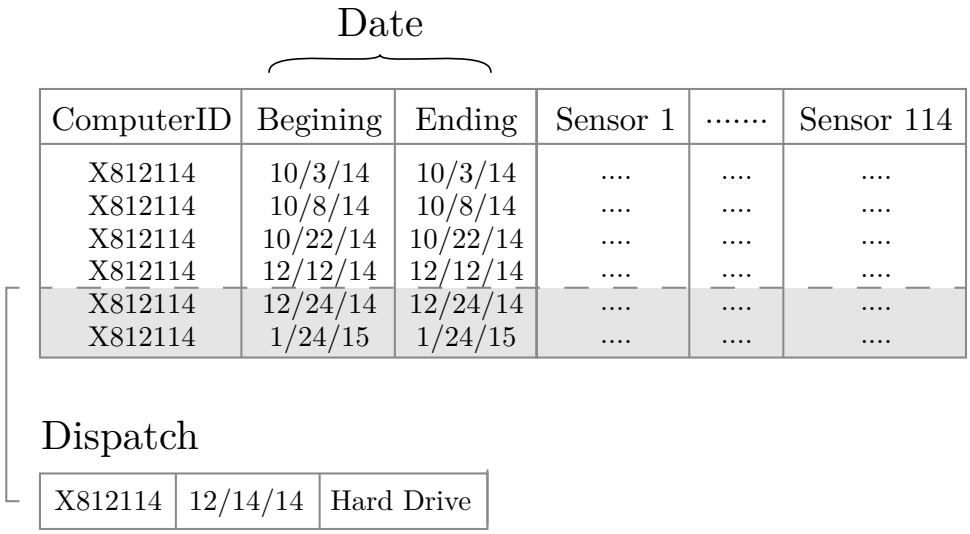


Figure 5-6: Process of forming a slice of data corresponding to a specific computer and then removing the sensor readings after the dispatch date. These sensor readings cannot be used for building the predictive model as they happened after the dispatch event.

5.5 Labeled training data

A predictive model will need features to be trained with. With the data selected and refined, features are now generated from the sensor readings. Features can be generated in a broad number of ways, and feature generation is a field unto itself. A few possibilities include using the readings themselves, using mathematical functions to modify the readings, and running specialized algorithms. Regardless of the methods used to generate these features, the user or program will end up with *labeled training data* files. Each row in this data file is a labeled example (hardware failure or not) for a specific computer system, and has these features as columns.

Traditionally, feature generation is a manual and time-consuming process. Though feature generation tools do exist, different tools offer different capabilities, and some popular data science toolkits don't offer any at all. In this particular case, two different feature generation methods were considered to develop the *labeled training data*.

The first method uses the deep feature synthesis algorithm, presented by Kanter and Veeramachaneni [4], to create relevant features for the model. Features created with this algorithm include applied functions to sensor readings. These functions are *means, sums, standard deviations*, and *maximum and minimum values*, among others. The output file is DFSFeatures.csv. In total, 650 different features were created with this algorithm. Table 5-7 shows the structure of the created table.

In the second method, Microsoft Excel was used to filter the unique Computer_IDs. We selected the first entry for every Computer_ID in the file, organized in descending order by time. This data set represents the first reading of data available for prediction. Features generated using these first readings are considered naïve features. The output file is NaïveFeatures.csv, which includes all sensor data, along with a label if the hardware failed as pictured in Table 5-8. Table 5.6 shows a description of the output files with the created features.

	Stage	Files	Size (rows)	Size (MB)
5	Labeled training data	• NaïveFeatures.csv • DFSFeatures.csv	• 12,483 • 12,483	• 5 • 40

Table 5.6: Feature table description.

DFS Features

ComputerID	Label	Feature 1	Feature 2	·····	Feature 50

Figure 5-7: Labeled training data table (DFSFeatures.csv) has the Computer_ID field, 650 engineered feature fields, and Label.

Naïve Features

ComputerID	Label	Sensor 1	Sensor 2	·····	Sensor 114

Figure 5-8: Labeled training data table (NaïveFeatures.csv) has the Computer_IDfield, 110 sensor fields, and Label.

5.6 Models and results

The last stage consists of employing a machine learning algorithm, which utilizes *labeled training data* in order to train a model. Machine learning is a broad topic in itself, and as previously mentioned, there exist a plethora of model-building algorithms and approaches, depending on the problem type.

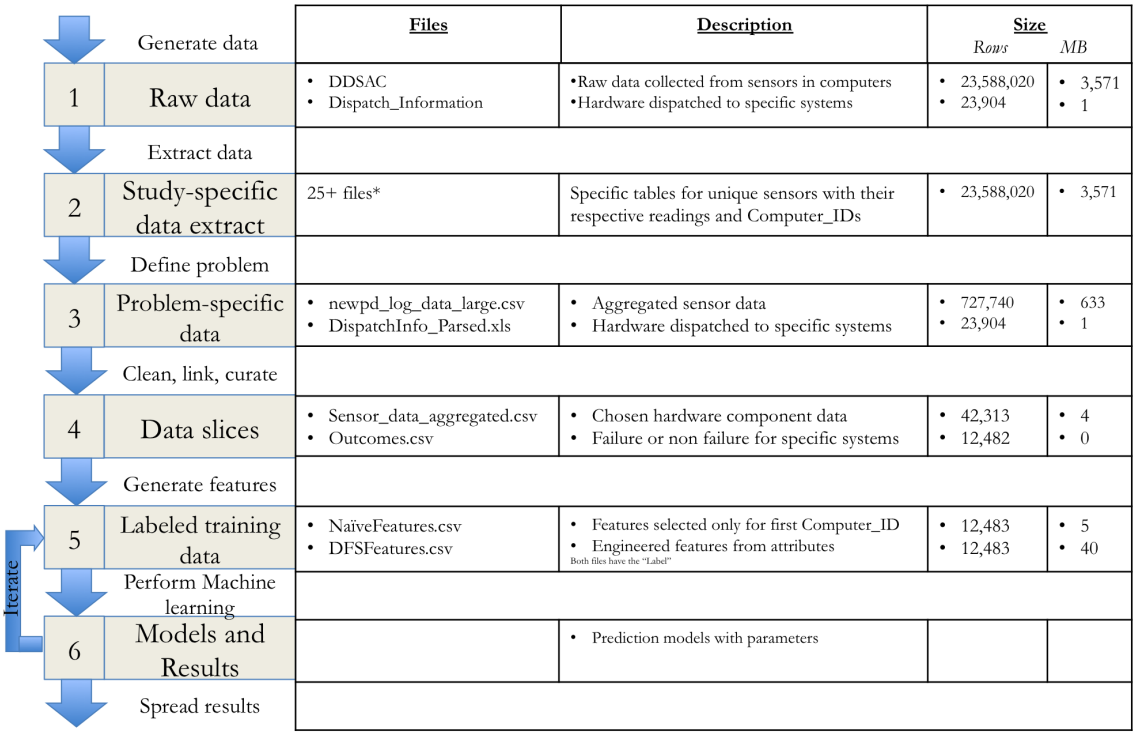
Once a model is trained, it is scored and evaluated according to its ability to predict. Depending on our model's accuracy (WA and FAR as explained in Chapter 4), we can continue iterating with other features and tuning algorithmic parameters until the desired accuracy is obtained.

Different tools were used to create these models. We first used FeatureLab, which generated a tenfold cross validated model that achieved a 0.72±0.04 WA (AUC).

The model used a Stochastic Gradient Descent (SGD) algorithm. Some of the most relevant features were:

- SUM(Readings.CPU_0_PCT)
- LAST(Readings.AC_Adapter_Type_W)
- MEAN(Readings.S4_mins)
- STD(Readings.Avg_ThreadCount)

Figure 5-9 shows a summary of the resulting files, tables, and characteristics that result from following data science process as previously described.



		Files	Description	Size Rows	Size MB
Generate data					
1	Raw data	• DDSAC • Dispatch_Information	•Raw data collected from sensors in computers •Hardware dispatched to specific systems	• 23,588,020 • 23,904	• 3,571 • 1
Extract data					
2	Study-specific data extract	25+ files*	Specific tables for unique sensors with their respective readings and Computer_IDs	• 23,588,020	• 3,571
Define problem					
3	Problem-specific data	• newpd_log_data_large.csv • DispatchInfo_Parsed.xls	• Aggregated sensor data • Hardware dispatched to specific systems	• 727,740 • 23,904	• 633 • 1
Clean, link, curate					
4	Data slices	• Sensor_data_aggregated.csv • Outcomes.csv	• Chosen hardware component data • Failure or non failure for specific systems	• 42,313 • 12,482	• 4 • 0
Generate features					
5	Labeled training data	• NaïveFeatures.csv • DFSFeatures.csv	• Features selected only for first Computer_ID • Engineered features from attributes Both files have the "Label"	• 12,483 • 12,483	• 5 • 40
Perform Machine learning					
6	Models and Results		• Prediction models with parameters		
Spread results					

Figure 5-9: Data science process and results. Note the reduction in size of data from 3.6GB to 40MB and 5MB by the time we bring it to a machine learning modeling method.

After finalizing the data science approach, we developed this proposed data science process, and carefully laid out the details to enable this process to be applied to any

prediction problem. Figure 5-10 shows the characteristics of the proposed end-to-end data science solution that arises from this process.

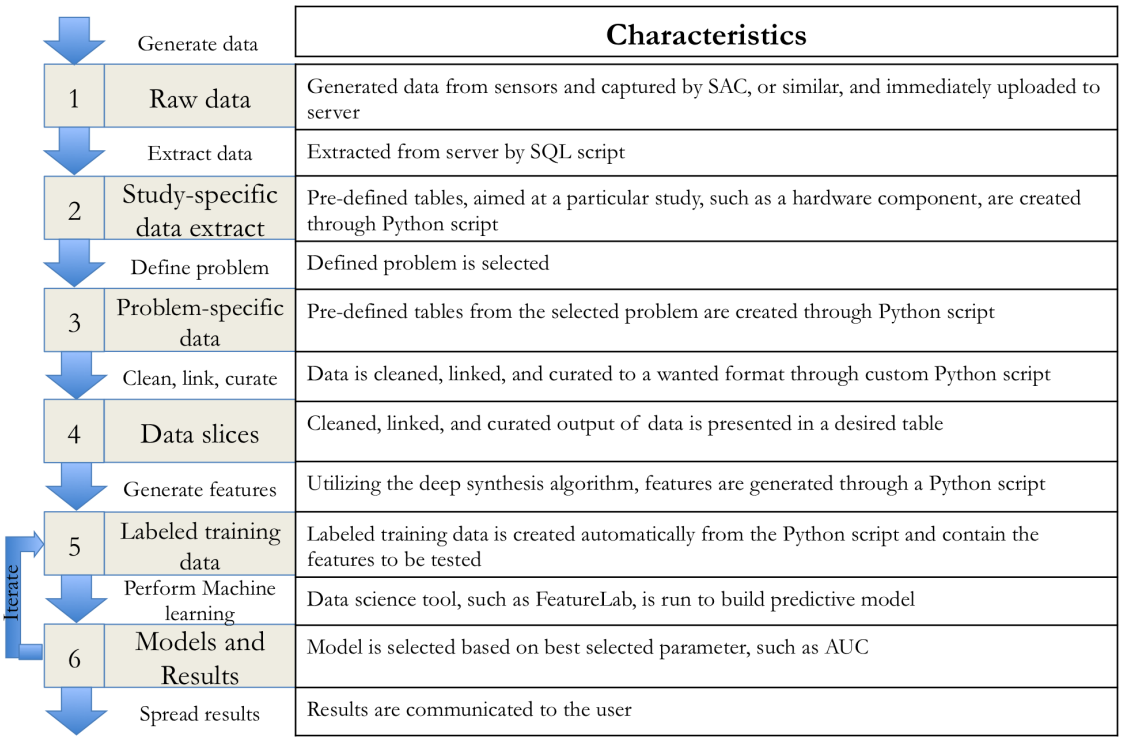


Figure 5-10: Proposed end-to-end data-driven solution.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 6

Machine Learning Tools

In this chapter, we utilize different platforms and software, referred to as "tools," to generate predictive models for our problem, while taking note of their differences in capabilities. We utilize Microsoft Azure, Amazon Machine Learning, IBM Watson Analytics, Nutonian Eureqa, BigML, and Skytree. Each of these platforms expect *labeled training data* as input[1]. The *labeled training data* are generated using two different methods:

1. We used FeatureLab's Deep Feature Synthesis (DFS) algorithm [4] to generate features from the sensor readings and form labeled training examples. We call these *DFSFeatures*.

2. We created features by following a *naïve* approach. We took the first available sensor readings for each *data slice* (see Section 5.4). We call these *NaïveFeatures*.

Training data summary: It is important to note that the data from the data slices is imbalanced, as there are 9,878 cases of "0" and 2,604 cases of "1." This means that 79% of the attributes in our target, "Label," are non-failures, and 21% are failures.

Goals: Each of these tools offer a variety of services for data in addition to machine learning and evaluation of the trained models. They could be broadly categorized into 5 functions:

[1]Except FeatureLab, which can generate features and could start with the *data slices* as input.

1. Ingestion and data visualization.
2. Data preparation.
3. Modeling and tuning parameters for training models.
4. Model evaluation.
5. Model analysis.

With our experiments, our core focus is on loading the data, training a machine learning model, and evaluating the trained model in terms of predictive accuracy. Thus we limit ourselves to stages two, three, and four. Our three goals are to evaluate:

1. Whether machine learning services provided by these tools can overcome the lack of feature engineering.
2. For the purposes of modeling and evaluation, what value these tools add compared to open-source tools like *scikit-learn* based machine learning software.
3. Which part of the end-to-end data science process these tools address.

What we are not evaluating:

1. Usability and User interface.
2. Input data format and size.
3. Model analysis.
4. Deployment functionality.
5. Language integration.

6.1 Ingesting and preparing the data

Each tool had different settings for uploading the *labeled training data* and manipulating it afterwards. The data preparation steps we performed in the specific tools are summarized in Table 6.1.

Table 6.1: Data ingestion and preparation. For all tools, *Labeled training data* was uploaded as a *csv* file. Some tools offered options to *select columns* and *features*, handle *missing values*, among others. This table shows specific steps we took in each tool to prepare the data for machine learning.

Tool	Availability	Data Preparation
Microsoft Azure Machine Learning	SaaS	-Drop columns with entire missing values -Missing values set to mean -Feature selection: Select top 45% of features using Pearson correlation between columns and label (target)
Amazon Machine Learning	SaaS	-Column selection possible -Modify column data type (Computer_ID to categorical, Label to Binary)
BigML	SaaS	-Reduced *DFSFeatures* to first 5,014 rows (40% of original size)[2]
IBM Watson	SaaS	-Used Azure to identify top 49 influential features using Pearson correlation between sensor data and label (target)[3]
Nutonian Eureqa	Desktop	-Normalize data[4] -Missing values set to mean
Skytree	Desktop	-None

[2]The tool imposed a constraint on the data size we could feed to it. This was because we were using a free version.

[3]The limit on number of features to be used was placed by this tool. However, no feature selection process or guidance was provided. Again, this limitation was perhaps for the free version we were using.

[4]For automatically-detected columns based on data; subtracted mean and divided by the standard deviation.

6.2 Modeling and choosing parameters

Once the *labeled training data* was uploaded and prepared, we then built predictive models by setting the *hyperparameters* as listed in Table 6.2.

Table 6.2: Modeling and parameter choices. The tools allowed different levels of customization for *hyperparameters*. Highlights: Skytree offered automatic modeling techniques and parameter tuning. IBM Watson did not allow setting parameters.

Tool	Modeling Technique	Parameter Settings
Azure ML	Decision tree	- Resampling method = *bagging* - Number of decision trees = 8 - Maximum depth of DT = 32 - Number of random splits per node = 128 - Minimum number samples per leaf node = 1
	Neural Network	- Number of hidden nodes: 100 - Learning rate = 0.1 - Number of learning iterations = 100 - Initial learning weights diameter = 0.1 - Momentum = 0 - Normalizer type = *MinMax*
Amazon ML	Logistic Regression	- Maximum ML Model size = 2000MB - Maximum data passes = 100 - Regularization = L2 - Regularization amount = 1e-4 - Data shuffle = *Auto*
BigML	Decision Tree	- Object weights 1 to 19 (Label 1) - Object weights 19 to 1 (Label 0) - Node threshold = 512
IBM Watson	Decision Tree	- Automatic
Nutonian	Symbolic Regression[5]	- Chose formula building blocks: Constant, Input variable, Addition, Subtraction, Multiplication, Division, Logistic function, Step function, If-then-else, Minimum, Maximum
Skytree	Gradient Boosted Tree	- Automodel (can chose: GBT, RDF, GLM, SVM). Skytree chose GBT[6] - Search iterations = 10 - Classifier testing; F-score = 0.1

[5]Symbolic regression is a specialized machine learning algorithm developed to identify non-linear functional relationships in the data.

[6]Model parameters selected by Skytree. Parameters for FirstSensorData: Number of trees = 58, Tree depth = 0, Max splits = 14, Learning rate = 0.104905, Regularization bins = 141, Probability threshold = 0.2152. Parameters for DFSFeatures: Number of trees = 234, Tree depth = 4, Learning rate = 0.126931, Regularization bins = 192, Probability threshold = 0.1984.

6.3 Evaluating the trained models

Next, we evaluated the trained models. They were evaluated against the data they were trained with (*train data*) and against the data they were not trained with (*test data)*. Different tools offered different options for the evaluations. Some offered the ability to split the data into train and test via a parameter. Some offered the ability to do $k - fold$ cross validation. The tools offered different settings as shown in Table 6.3.

Evaluation metrics The trained predictive models were evaluated with different metrics, which were:

1. **AUC (Area under the curve)**

   The AUC is a way to measure the accuracy of the model. As seen in Figure 6-1, it is the total area under the ROC (receiver operating characteristic) curve. The ROC curve is plotted on a graph that represents the false-positive rate on the x-axis and the true-positive rate on the y-axis. It means that the area measures discrimination, which is the ability of the model to correctly classify one category against another.

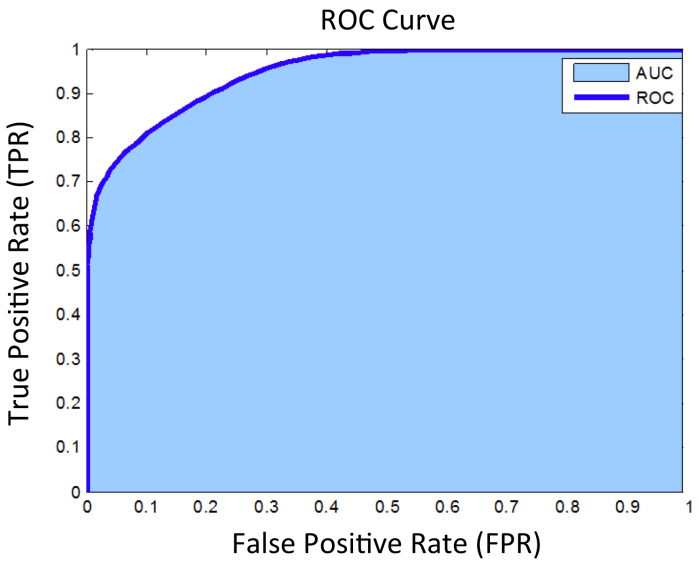


Figure 6-1: The AUC metric is the area under the ROC curve.

2. **FPR (False-positive rate)**: The FPR is the percentage of examples belonging to the negative class that have been incorrectly classified as a positive result. It is calculated with the following equation: $\frac{FP}{(FP+TN)}$, where FP is the number of "false-positives" and TN is "true-negatives."

3. **Precision**: This is also known as the "positive predictive value." It is calculated with the following equation: $\frac{TP}{(TP+FP)}$, where TP is the number of "true-positives" and FP is the number of "false-positives."

4. **Recall**: Recall is also known as the "true-positive rate." It is calculated with the following equation: $\frac{TP}{(TP+FN)}$, where TP is the number of "true-positives" and FN is the number of "false-negatives."

Table 6.3: This table shows the evaluation techniques, which included cross validation for some tools, as well as splitting the data into *train* data and *test* data for building and evaluating the predictive models. Data was always split as: 70% for training and 30% for testing. Only three tools - Nutonian, Skytree, and Azure offered ability to do cross validation.

Tool	Evaluation technique
Microsoft Azure Machine Learning	- Cross-validation with 10 folds - Splitting*
Amazon Machine Learning	- Splitting**
BigML	- Splitting
IBM Watson	- Automatic**
Nutonian	- Automatic cross-validation - Splitting*
Skytree	- Cross-validation with 10 folds - Splitting*

* Offers option to change split settings
** Does not offer option to change split settings

6.4 Results

Each predictive model was evaluated against these four particular metrics. The models were evaluated against the *train data* (70% of the data) and the *test data* (30% of the data), as shown in Table 6.4 and Table 6.5.

Table 6.4: Performance of different tools for our problem - NaïveFeatures. The predictive models created by each tool were evaluated with different datasets for their performance. The datasets were *train data*, which was comprised of 70% of the data in the original *NaïveFeatures* and *test data*, which was comprised of 30% of the data in the original *NaïveFeatures* and was not used to train the model. Some tools were evaluated with 100% of the data since they did not offer the option to split the data, as seen in table 6.3.

		Train data - 70%				**Test data- 30%**			
Tool	**Algorithm**	**AUC**	**FPR**	**Precision**	**Recall**	**AUC**	**FPR**	**Precision**	**Recall**
Microsoft Azure Machine Learning	Decision Tree	0.639	0.010	0.365	0.027	0.641	0.010	0.426	0.037
	Neural Network	0.617	0.011	0.462	0.046	0.629	0.010	0.486	0.046
Amazon Machine Learning	Logistic Regression	0.997	0.010	0.961	0.932	0.610	0.010	0.431	0.028
BigML*	Decision Tree	0.770**	0.011	0.866	0.272	0.592**	0.020	0.263	0.027
Skytree GUI	Gradient Boosted Trees	0.822	0.014	0.810	0.235	0.826	0.012	0.844	0.238

		All data - 100%			
		AUC	**FPR**	**Precision**	**Recall**
IBM Watson Analytics	CHAID Decision Tree		0.442	0.284	0.664
Nutonian Eureqa	Symbolic Regression	0.568**	0.015		0.050

* Estimated metrics based on creating an ROC curve from the confusion matrices by varying the label weights

** Graphically estimated

Table 6.5: Performance of different tools for our problem - DFSFeatures. The predictive models created by each tool were evaluated with different datasets for their performance. The datasets were *train data*, which was comprised of 70% of the data in the original *DFSFeatures* and *test data*, which was comprised of 30% of the data in the original *DFSFeatures* and was not used to train the model. Some tools were evaluated with 100% of the data since they did not offer the option to split the data, as seen in table 6.3.

		Train data - 70%				**Test data- 30%**			
Tool	**Algorithm**	**AUC**	**FPR**	**Precision**	**Recall**	**AUC**	**FPR**	**Precision**	**Recall**
Microsoft Azure Machine Learning	Decision Tree	0.708	0.011	0.647	0.096	0.688	0.013	0.581	0.087
	Neural Network	0.674	0.009	0.637	0.075	0.674	0.011	0.619	0.084
Amazon Machine Learning	Logistic Regression	0.999	0.010	0.963	0.996	0.680	0.010	0.646	0.067
BigML*	Decision Tree	0.928**	0.003	0.977	0.410	0.611**	0.067	0.391	0.175
Skytree GUI	Gradient Boosted Trees	0.974	0.015	0.935	0.814	0.980	0.014	0.944	0.843

		All data - 100%			
		AUC	**FPR**	**Precision**	**Recall**
IBM Watson Analytics	CHAID Decision Tree		0.3181	0.623	0.346
Nutonian Eureqa	Symbolic Regression	0.659**	0.015		0.098

* Estimated metrics based on creating an ROC curve from the confusion matrices by varying the label weights

** Graphically estimated

6.5 Key findings

Having utilized these different machine learning tools, several notable findings emerge. We divide these findings into two sets: the first provides insights about the model development process, while the second provides an evaluation of the machine learning components of these tools.

Model development process

1. **Feature engineering provides a significant boost in predictive accuracy.** Features developed through the DFS algorithm provided valuable information to create more accurate predictive models[7]. When using *DFSFeatures* as the *labeled training data*, there was a notable improvement in the AUC for the ROC curve. As seen in Table 6.6, feature engineering increased AUC by an average of 13% when models were evaluated on *train data*, and by an average of 11% for models evaluated with *test data*. Arguably, since the feature generation process used multiple sensor readings rather than just the first available sensor readings (as in naïve features), this finding may be *intuitive*.

Table 6.6: AUC *DFSFeatures* vs. AUC *NaïveFeatures* shows a vast improvement in the models' predicting accuracy when using *DFSFeatures* instead of the *NaïveFeatures*.

AUC improvement		
	Train data - 70%	**Test data - 30%**
Azure ML - Decision Tree	10.8%	7.3%
Azure ML - Neural Network	9.2%	7.2%
Amazon ML	0.2%	11.5%
BigML	20.5%	3.2%
Skytree	19.7%	18.7%
Average	**12.1%**	**9.6%**
	All data - 100%	
Nutonian Eureqa	16.1%	16.1%
Total Average	**12.8%**	**10.7%**

2. **Model selection and hyperparameter tuning cannot overcome the lack of feature engineering.** Of all the tools, only Skytree offered the ability to

[7] Although we used the deep feature synthesis algorithm in this thesis, we envision similar benefits could be achieved with manual feature engineering.

automatically select models and tune their *hyperparameters* as seen in Table 6.2. The "automodel" option selected a Gradient Boosting Tree algorithm to train the model out of the available classification algorithms[8]. With *NaïveFeatures* it was able to generate a predictive model with an AUC of 0.83 (on *test data*, refer to Table 6.4), a *recall* of 0.24, at a *FPR* of 0.01, while other tools, even with the *DFSFeatures*, were only able to achieve a 0.72 AUC[9] as shown in Table 6.5. However, when Skytree's tool was given *DFSFeatures* it performed *even better*, achieving a 0.98 AUC and a 0.84 *recall* at a *FPR* of 0.01. This means an increase of 19% in the AUC and a 254% boost in *recall*, while maintaining an *FPR* of 0.01. Hence, while model tuning can enable better accuracies even from simple *NaïveFeatures*, there is a possibility that *engineered* features can enhance the accuracy and evaluation metrics even further.

Evaluation of the tools

1. **All these tools combined only solve one small stage of the data science endeavor.** All of the machine learning tools needed the input of *labeled training data* to start training a machine learning model[10]. Therefore, these tools can only be used after the completion of the first four stages of the data science process (described in Chapter 5 and shown in Figure 5-10). A majority of our time was spent in these first four stages, and we imagine that most data science endeavors are similar. Further, if we wish to change the prediction problem from predicting *hard drive* failures to *motherboard* failures, much of the work has to be done outside of these tools. Arguably, a tool that can enter the process earlier will provide more flexibility and agility.

2. **A few lines of python code can replace the core machine learning functionality provided by these tools.** A few lines of code written using a popular package in python called *scikit-learn* can perform most of the modeling functions these machine learning tools offer. These include *splitting* the data into

[8]Skytree offered the option to chose between GBT, RDF, GLM, and SVM algorithms.
[9]0.72 AUC was achieved using FeatureLab's modeling method as reported in section 5.6.
[10]Except FeatureLab, which can work with *data slices* and generate *labeled training data*.

train and *data* sets, performing *cross validation*, *training* a model, and *testing* a model. Below we provide a simple code[11] snippet that, when provided a feature matrix in X and target label in y can perform all the functions mentioned here.

```
from sklearn import svm
from sklearn.cross_validation import cross_val_score
from sklearn.metrics import roc_auc_score
from sklearn.cross_validation import train_test_split

#Splitting the data into test and train
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size
    =0.33)


#Choose a classifier and its hyperparameters
clf = svm.SVC(kernel='linear', C=1)

# Perform cross validation
scores = cross_val_score(clf, X_train, y_train, cv=10, scoring="
    roc_auc", n_jobs=-1)
cv_auc = scores.mean()

# Learn a classifier via fit method
clf.fit(X_train, y_train)

# Test the classifier
y_predict = clf.predict(X_test)
print roc_auc_score(y_test, y_predict)
```

3. **The tools provided other important services.** However, these tools offered additional services. One consistent service among all SaaS-based offerings was persistent storage of data and models, allowing a user with an account to login and reuse the models that s/he previously trained. Additionally, these tools presented intuitive graphical user interfaces, and enabled easy evaluation and

[11]The code snippet was provided by Max Kanter.

execution of models on new data. However, the tools varied a lot in visualization, exploration, and analysis techniques, both for data and models.

4. **The tools varied in how they evaluated models.** There is general agreement in the machine learning community about how to train and evaluate classification models, and it was surprising to see that these tools did not follow such standards. Some of them did not offer a capability to cross validate models. Some did not allow the user to specify the train/test split. Others did not allow splitting the data, and expected that the users do that outside the tool. This is described in Table 6.3. These variations made it harder to compare the accuracy of solutions that were developed on them.

5. **For most predictive problems, the size of the *labeled training data* will likely be *small*. This allows for training a model on a desktop.** Arguably, these tools were developed for large/big data sets and offer scalable approaches for training models. Hence, the tools cannot be compared to an open source machine learning software like the one above, since the previous machine learning software does not provide immediate scaling. However, in our experience, while the *raw data* and the *study-specific data* comprised whole gigabytes, by the time we selected the prediction problem, generated the features, and assembled the *labeled training data*, the data size shrank to a couple of megabytes, as seen in Figure 5-9. Data size aside, we only had 12,000 training examples - a dataset for which we could train models even on an off-the-shelf computer.

THIS PAGE INTENTIONALLY LEFT BLANK

Chapter 7

Financial Impact Analysis of the Proposed End-To-End Data-Driven Solution

The potential to deploy an integrated end-to-end solution for hardware failure prevention can reap tangible monetary benefits for Dell as well as benefits such as better brand reputation and the providing of a better customer experience. Moreover, Dell's suppliers are also substantially impacted by this problem. In order to quantify the financial opportunities, the following two business cases are presented with a conservative approach:

1. The first case encompasses Dell and the hardware component suppliers.

2. The second case only considers Dell.

7.1 Incurred costs

Dell and the hardware suppliers currently have incurred costs of manufacturing or buying, storing, shipping failed hardware components to be replaced, attending to customers' calls and requests, among others. As mentioned, two business cases are developed to estimate the incurred costs from hardware failures.

In the first case, a moderate approach is taken and only two relevant costs are considered. Theses costs that are taken into account are replacing the physical hardware components and handling the calls related to the failures. The weighted average hardware component cost for the analyzed laptops and desktops is assumed to be $34. "Laptops and desktops" will be referred to as "systems." Based on the extracted and cleaned data, systems have an 8.2% yearly failure rate, but 33% of these failures are considered to be due to accidents [30]. Therefore, only 5.5% is the "effective" failure rate for hardware components. The user base consideration for this case is composed of two segments: new users and previous users. The new user base accounted accords with the deployments of new systems, which amounts to 18.8M new systems per year. From the previous user base only an estimated 30% of users are accounted for from the two previous years, as they are considered to have bought an extended warranty [31], which usually lasts three years. This brings the active user base to a total of 30M+ users. The hardware manufacturers, including Dell, cover all these systems under a warranty. Utilizing the previous numbers, the calculated cost for all hardware component suppliers exceeds $56M per year for failed hardware components. Another important cost in this case is attending to customer calls. At a calculated rate of 15M calls per year, an average call time of 25min, and a conservative cost of $17 per hour per call [32], the total cost for attending to these calls from failures exceeds $107M per year. Therefore, as seen in Figure 7-1, the overall estimated expense to Dell and their suppliers is approximately $163M per year for hardware failure components and handling the respective calls.

On the second case, Dell covers batteries under warranty. The assumed battery cost is $25. The failure rate is considered to be 1.9% of hardware failures, which means that systems fail 0.16% of the time due to a battery failure. Also taking into account that 33% of the failures are due to accidents, the considered "effective" failure rate is 0.103%. This case considers the same user base for new users and previous users as the previous case, but also considers a mix of 50% laptops and the rest desktops, making the active user base 15M+ users. With the previous numbers and assumptions, the incurred hardware costs to Dell for battery failures are estimated at $0.4M per

year. Also, the more relevant cost component is the customer service to attend to calls when there is a battery failure. Considering the same assumptions for the call center as before, but considering only 0.3M calls for batteries, the cost to handle these failures is approximately $1M per year. Hence, the hardware and customer service costs for Dell to deal with battery failures are estimated to be $1.4M per year, as seen in Figure 7-1.

7.2 Investment

The investment necessary to develop and deploy the data driven tool is estimated to be between $0.75M and $2.1M, depending on the reach of the process and service. These amounts are based on previous project experience and taking into account the required tasks and time to complete them by working function. Most of the investment cost consists in labor costs. As seen before, the technology to integrate this solution is already available on the market and could be utilized at very low cost.

7.3 Impact

The proposed end-to-end solution can be utilized to effectively monitor the hardware components in a system and successfully predict failures with high WAs. In this case, the developed models with FeatureLab provide an accuracy of 70%+ in correctly predicting the failure of a hard drive, which is the main hardware component that fails. For the impact analysis, the accuracy of the developed model is extrapolated to the rest of the components. Taking a conservative approach, the prevention of hardware failing is assumed to have an overall 50% effectiveness, meaning that half of the failures that were predicted could be prevented with the proposed end-to-end solution. This assumption is approximately 70% of the predictions that the current and most optimized model in FeatureLab could achieve, which was 72%. This prevention will also reduce by 50% the number of calls for these failed components. For the first case, where all the suppliers and Dell are considered since all components are covered

by a warranty, the impact of the expected yearly savings on hardware components would be $28M, the customer service calls avoided would amount to $54M, and this would total to approximately $407M for the net savings in a five-year period. For the second case, where batteries are covered by Dell's warranty, the yearly hardware savings would reach $0.2M, the customer service calls avoided for batteries would be $0.5M, and approximately $2.7M could be obtained in net savings during the first five years. Figure 7-1 shows the yearly impact if the process to prevent failures were as accurate as the model. This described impact is a conservative approach to the potential that this solution can have in hardware and customer service cost savings. More realistically, additional savings need to be considered in supply chain, storage costs, inventories, among others.

7.4 Going big

The impact this end-to-end solution can have around the globe for this specific use case is worth considering. IDC forecasts shipments of 101M consumer desktops and 40M consumer laptops for 2016 and cites a previous user base of 209M consumer desktops and 89M consumer laptops from the past two years [33]. Taking into account that 30% of previous users have extended warranties, the active user base totals 230M+. An average hardware component's cost is also assumed to be $34. Failure rates are considered the same, at 8.2% system failures per year, and with an accident rate of 30% leaves an effective failure rate of 5.5%. Considering hardware costs, the total impact hardware failures have around the globe exceeds $431M per year. Additionally, if we consider the same user base, same competitive call center costs, and a conservative 30% call rate for hardware failures globally [34] [35], the impact exceeds $490M. Therefore, the worldwide impact exceeds $920M per year on hardware and customer service calls. Utilizing the 72% accurate prediction model created with FeatureLab, the value of this solution for this specific case surpasses $660M per year. As previously mentioned, this does not take into account any saved costs in inventories, supply chain, working capital, other support systems, among others.

The financial impact that hardware failures have on manufacturers and service providers is of important consideration. There are definitely other considerable scenarios, where the players' involvement and the solution's scope varies, such as considering that an implementation of this predictive solution would reduce calls to Dell across all hardware components, not just batteries, and have a much bigger impact. These different scenarios could be constructed from the mentioned assumptions and from Figure 7-1, which demonstrates the impact of the results of this study to the different players.

Financial Impact Results Summary				
Assumptions		**Global**	**Dell and Suppliers**	**Dell**
Userbase	users	230,547,848	30,008,000	15,004,000
Hardware Cost	/ component	$34	$34	$25
Effective Failures	%	5.49	5.49	0.10
Cost	/ year	$430,532,526	$56,037,912	$384,505
Call rate	/ year	69,164,354	15,066,229	142,765
Average call time	min / call	25	25	25
Call cost	/ hr	17	17	17
Cost	/ year	$489,914,177	$106,719,121	$1,011,249
Hardware Failure Prevention		$430,532,526	$56,037,912	$384,505
Call Center Call Reduction		$489,914,177	$106,719,121	$1,011,249
Total Impact		$920,446,703	$162,757,033	$1,395,754
Considering Model Performance of 0.72				
Hardware Failure Prevention		$309,983,419	$40,347,297	$276,844
Call Center Call Reduction		$352,738,207	$76,837,767	$728,099
Total Impact		$662,721,627	$117,185,064	$1,004,943

Figure 7-1: Summary of the financial yearly implications for both use cases and an estimated global impact. Different variations of these use cases can be built using the given information.

Finally, a sensitivity analysis was developed to understand the variability on the financial impact depending on the accuracy of the model. The results show an important impact depending on the use case and the industrial player. In the first case, a 10% variability in the model to prevent hardware failures would impact Dell and their suppliers in $16M per year. In the second case, the impact of a 10% variability in the model would impact Dell in $0.1M per year. The biggest influence is the call center cost. These results for the models effectiveness are summarized in Figure 7-2.

Sensitivity Analysis							
Model Effectiveness (AUC)	**20%**	**40%**	**60%**	**80%**	**100%**	*10% Δ in effectivness*	*% of impact*
Global Impact							
Hardware Failure Prevention	$86,106,505	$172,213,011	$258,319,516	$344,426,021	$430,532,526	*$43,053,253*	*47%*
Call Center Call Reduction	$97,982,835	$195,965,671	$293,948,506	$391,931,342	$489,914,177	*$48,991,418*	*53%*
Total Impact	$184,089,341	$368,178,681	$552,268,022	$736,357,363	$920,446,703	*$92,044,670*	*100%*
Dell and Suppliers Impact							
Hardware Failure Prevention	$11,207,582	$22,415,165	$33,622,747	$44,830,330	$56,037,912	*$5,603,791*	*34%*
Call Center Call Reduction	$21,343,824	$42,687,648	$64,031,472	$85,375,297	$106,719,121	*$10,671,912*	*66%*
Total Impact	$32,551,407	$65,102,813	$97,654,220	$130,205,626	$162,757,033	*$16,275,703*	*100%*
Dell Impact							
Hardware Failure Prevention	$76,901	$153,802	$230,703	$307,604	$384,505	*$38,451*	*28%*
Call Center Call Reduction	$202,250	$404,500	$606,749	$808,999	$1,011,249	*$101,125*	*72%*
Total Impact	$279,151	$558,302	$837,453	$1,116,603	$1,395,754	*$139,575*	*100%*

Figure 7-2: Sensitivity analysis of the impact of the model's effectiveness. The call center's call reduction impact is very relevant to Dell's case. Generally, it is worth noting how the performance of the predictive model has a very important financial impact. Hence, the relevance of the data, generated features, modeling techniques and tuning settings.

Chapter 8

Conclusions

The analysis, results, and potential for implementation of an end-to-end solution demonstrate through this work an interesting potential for the proposed data science process and prevention of hardware failure through sensor data. Dell has many of the necessary tools and capabilities to implement such solutions.

8.1 Key Findings

The findings of this work can be explained in two areas: prevention of hardware failure at Dell and evaluation of machine learning tools.

Prevention of hardware failure at Dell

The "quality of a model is as good as the data used in its construction," [36] and so far this work has presented a potential solution with promise for a real and tangible impact for Dell, its customers, and its suppliers. Taking the lessons and observations from the current prevention system and the proposed solution, an improved end-to-end solution would require the following characteristics:

1. **Real-time integration of SAC**

   Data is currently stored in summary files in each system and mostly only extracted when a failure happens. The change needs to be implemented in the following two ways:

(a) Real-time extraction of data from the systems' hardware sensors that would be collected on to the server.

(b) Compressed storage of historical data from systems that is uploaded at defined dates.

2. **Real-time integration of new data to constructed models**
   As new data is imported from systems, the previously created models can be re-evaluated for improved prediction accuracy.

3. **Real-time feedback to the user**
   Models can be exported to a program in the users' system (such as SAC), or run on the cloud, which would be used to monitor for potential hardware failures with high accuracy and fast response time to the user.

Additional work with more historical data needs to be done in order to improve the developed prediction models for hard drive failures. One of the most limiting findings, while selecting the data, was that any failure that was going to be analyzed had no more history than the current log for the day when it failed. This was a limiting factor to the insight that could be obtained from this data.

Machine learning tools

The utilization of the different machine learning tools enabled us to compare their functionalities. This makes these tools particularly dependant on the required uses and goals. Among the tools, this study demonstrated the following findings:

1. Varied predictive modeling performances, with non-standardized metrics, and different visualization options.

2. The tools only solve the last stage of the data science process, needing much manual work in the earlier stages.

3. These tools are ready to scale, as they offer additional services such as data and model storage and updating.

4. The DFS algorithm demonstrated to have improved the accuracy of the built machine learning models against using a traditional naïve approach. This method to use DFS-generated features translates into added value for solving prediction problems

The machine learning tools provide a user-friendly and intuitive platform with easy access for anyone to become a "citizen data scientist." However, the tools showed varied performance and functionalities and it is important for the user to know their modeling, metrics, and visualization requirements when choosing one of these tools.

8.2 Contributions

This work proposes new concepts in the subject of AI, a data-driven solution, the prevention of hardware failure, and improving the performance of predictive models. The contributions of this work are summarized in the following statements:

1. Recommended an **AI taxonomy and framework** to facilitate the understanding of AI as a tool and as a strategy in the pragmatic business sense as "Smart Machines." The taxonomy divided AI into the following technologies: machine learning, deep learning, image recognition, NLP and NLI, and prescriptive analytics. The taxonomy represented AI in three layers: smart infrastructure, smart data, and smart apps and services.

2. A **competitive analysis of the different AI technologies** and business applications was presented as well, as an estimation for the Smart Machines market. This showed potential opportunities specifically for machine learning applications.

3. This thesis proposes a **data science process** with six stages to follow when there is a goal to predict a specific occurrence. The specific stages to complete are: raw data, study-specific data extract, problem-specific data, data slices, labeled training data, and models and results.

4. We **compared seven different machine learning tools** and their predictive models for the particular problem of hard drive failures. The most accurate model with *test data* (out of sample data) was achieved using *DFSFeatures* by Skytree with an AUC of 0.98 and a *recall* of 0.84 at a *FPR* of 0.01.

5. We demonstrated that **DFS-generated training data improved the performance** of the built machine learning models. This increased the predictive accuracy, measured by the AUC, on an average of 13% for *train data* and an average of 11% for *test data*.

6. This work presents a **new approach to prevent hard drive failures** through data from sensors and following the proposed the data science process.

8.3 Recommendations

Through this analysis, it is apparent that Dell has the necessary tools to implement such end-to-end data science solution in order to have a real impact in their business and their customers' experience. Additionally, once the impact of a desired technology or prediction problem is understood, the proposed data science process can be followed for the development of new use cases. This data-driven process is transferable to other parts of the business and products.

8.4 Future projects

There exist 16 new concepts that were developed during a brainstorming session at Dell, and three more that a detailed financial and technological analysis was performed. There are four immediate alternatives that can be pursued at Dell are:

1. Implementation of the proposed end-to-end solution for hardware failure prevention.

2. Further data validation of ideas, in particular the other top three concepts that were identified in:

- Security
- Serviceability
- Productivity

3. Research additional Smart Machines opportunities for Dell's potential entrance in this market.

4. Identify a new prediction problem of interest and follow the proposed data science process to develop a new particular end-to-end solution.

8.5 Conclusions

AI is no longer a far away futuristic field, but a real and tangible day-to-day technology that we are utilizing to enhance processes, business, and our daily lives. The amount of progress and new tools that have been developed during the past year has been fascinating. As seen throughout this work, Smart Machines and the data science process are a very applicable concept to businesses and there is great potential for a plethora of use cases. A great deal of technology is easily available through automated platforms and even open source software that are reducing the entry-barriers for new players that can disrupt the traditional businesses. However, there are many variations on the functionalities of the available machine learning tools and software. But very importantly, these tools provide realistic improvements to conventional processes and are a powerful alternative that can make of everyone a *citizen data scientist*.

After having taken a deep dive into the wide, complex, and amazing field of AI, there was a great deal of learning that provided an entrance into a field that is changing the world as we know it. Utilizing the same core technology, such as machine learning, and following the proposed data science process, there is much opportunity for implementations in a variety of cases. These cases are very relevant to many areas such as preventive maintenance of all industries, financial applications, disease treatment and prevention in health care, prediction of the outcome for any process, among others. Dell and any business have many detected opportunities, and

this detailed analysis was for just one business case. There are many more uses that eagerly wait to be explored around us with this data science process and machine learning tools.

Appendix A

AI applicable concepts

		Parameters					
		Tech-readiness (feasibility / ease of implementation) 1 - 4 5 yrs - ready	Return / Impact * Financial 1 - 4 low - high (20+%)	Return / Impact * Reputation 1 - 4 low - high	Return / Impact * IP 1 - 4 low - high	CS/Dell Fit (Business Alignment) 1 - 4 low - high	Total 1 - 4 low - high
Security	**Idea**						
	Automated Data Classification						
	Advanced Threat Protection						
	Data Behavior						
	User Behavior Classification						
	Environment & Contextual Security						
	Privacy Advisor						
Serviceability	**Idea**						
	User self-help Q&A						
	Predictive Analytics (Failure Suggestions)						
	Software (IT Rot)						
	Call/ Conversation Sentiment						
	Recommendation Engine						
Manageability	**Idea**						
	Automation						
	Self-management & healing						
	Schedule Optimization						
	Who are you gonna call?						
	Patch Advisor						
Productivity	**Idea**						
	Dynamic Profile						
	Know-me (self-learning)						
	Kown-me better (context, env. Aware)						
	Pre-process / visualize data						
	Monitor-suggest Data Reports						
	Personal Productivity Enhancer						
	Automated email classification						
	Disruption filter						
	Scheduling Meetings (smart collaborator -rooms)						
	VAs (Sentiment analysis)						
	Individual productivity Awareness						

THIS PAGE INTENTIONALLY LEFT BLANK

Bibliography

[1] K. F. Brant and T. Austin, "Hype Cycle for Smart Machines, 2015," Gartner, Tech. Rep., Jul. 2015. [Online]. Available: http://www.gartner.com/document/3099920?ref=solrAll&refval=162870108&qid=74f1fe050fe3befc42150d6ba0d1149a

[2] "June Oven." [Online]. Available: https://juneoven.com

[3] D. Pierce, "This Smart Oven Bakes Perfect Cookies Without Your Help," Jun. 2015. [Online]. Available: http://www.wired.com/2015/06/june-oven/

[4] J. M. Kanter and K. Veeramachaneni, "Deep feature synthesis: Towards automating data science endeavors," in *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on.* IEEE, 2015, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=7344858

[5] M. J. Kane, "Cleveland's action plan and the development of data science over the last 12 years," *Statistical Analysis & Data Mining*, vol. 7, no. 6, pp. 423–424, Dec. 2014. [Online]. Available: http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=a9h&AN=99708587&site=eds-live

[6] T. Urban, "The Artificial Intelligence Revolution: Part 1 - Wait But Why," Jan. 2015. [Online]. Available: http://waitbutwhy.com/2015/01/artificial-intelligence-revolution-1.html

[7] C. Smith, B. McGuire, T. Huang, and G. Yang, "The History of Artificial Intelligence," University of Washington, Tech. Rep., Dec. 2006. [Online]. Available: http://courses.cs.washington.edu/courses/csep590/06au/projects/history-ai.pdf

[8] "IBM - What is big data?" [Online]. Available: http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html

[9] R. Viereckl, D. Ahlemann, A. Koster, and S. Jursch, "Connected Car Study 2015: Racing ahead with autonomous cars and digital innovation," Strategy&, Tech. Rep., Sep. 2015. [Online]. Available: http://www.strategyand.pwc.com/reports/connected-car-2015-study

[10] L. Mearian, "World's data will grow by 50x in next decade, IDC study predicts," Jun. 2011. [Online]. Available: http://www.computerworld.com/article/2509588/data-center/world-s-data-will-grow-by-50x-in-next-decade--idc-study-predicts.html

[11] M. Mackay, "The Future of Artificial Intelligence," Apr. 2009. [Online]. Available: http://www.bit-tech.net/bits/2009/04/29/the-future-of-artificial-intelligence/1

[12] R. Kurzweil, "The Law of Accelerating Returns," Mar. 2001. [Online]. Available: http://www.kurzweilai.net/the-law-of-accelerating-returns

[13] J. Hagel, J. S. Brown, T. Samoylova, and M. Lui, "From exponential technologies to exponential innovation," Deloitte Center for the Edge, Tech. Rep. 2, 2013. [Online]. Available: http://www2.deloitte.com/content/dam/Deloitte/es/Documents/sector-publico/Deloitte_ES_Sector-Publico_From-exponential-technologies-to-exponential-innovation.pdf

[14] D. W. Cearly, M. J. Walker, and M. Blosch, "The Top 10 Strategic Technology Trends for 2015," Gartner, Tech. Rep., Jan. 2015. [Online]. Available: http://www.gartner.com/document/2964518?ref=feed

[15] "Machine Learning Applications for High Performance Computing - NVIDIA." [Online]. Available: http://www.nvidia.com/object/machine-learning.html

[16] D. W. Cearly, M. J. Walker, and B. Burke, "Top 10 Strategic Technology Trends for 2016: At a Glance," Gartner, Tech. Rep., Oct. 2015. [Online]. Available: https://www.t-systems.com/news-media/gartner-top-10-strategic-technology-trends-for-2016-at-a-glance/1406184_1/blobBinary/gartner_top10.pdf

[17] R. van der Meulen, "Why Should CIOs Consider Advanced Analytics?" Dec. 2104. [Online]. Available: http://www.gartner.com/newsroom/id/2950317

[18] "Artificial Intelligence. Landscape Report and Data." [Online]. Available: https://www.venturescanner.com/artificial-intelligence

[19] "Dell - Our history." [Online]. Available: http://www.dell.com/learn/us/en/vn/our-history

[20] M. J. De La Merced, "Dell to Buy EMC in Biggest Tech Takeover, a Year in the Making," *The New York Times*, Oct. 2015. [Online]. Available: http://www.nytimes.com/2015/10/13/business/dealbook/dell-announces-purchase-of-emc-for-67-billion.html

[21] G. Hamerly, C. Elkan, and others, "Bayesian approaches to failure prediction for disk drives," in *ICML*. Citeseer, 2001, pp. 202–209. [Online]. Available: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.89.3006&rep=rep1&type=pdf

[22] G. Hughes, J. Murray, K. Kreutz-Delgado, and C. Elkan, "Improved disk-drive failure warnings," *IEEE Transactions on Reliability*, vol. 51, no. 3, pp. 350–357, Sep. 2002.

[23] J. Murray, G. Hughes, K. Kreutz-Delgado, and D. Schuurmans, "Machine Learning Methods for Predicting Failures in Hard Drives: A Multiple-Instance Application," *Journal of Machine Learning Research*, vol. 6, no. 5, pp. 783–816, 2005. [Online]. Available: http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=aci&AN=18004064&site=eds-live

[24] P. Cheeseman and J. Stutz, "chapter Bayesian Classification (AutoClass)," in *Advances in Knowledge Discovery and Data Mining*. Menlo Park, CA: AAAI Press, 1995, pp. 158–180.

[25] F. Salfner, M. Lenk, and M. Malek, "A Survey of Online Failure Prediction Methods," *ACM Computing Surveys*, vol. 42, no. 3, pp. 10:1–10:42, Mar. 2010. [Online]. Available: http://libproxy.mit.edu/login?url=https://search.ebscohost.com/login.aspx?direct=true&AuthType=cookie,sso,ip,uid&db=bth&AN=49069494&site=eds-live

[26] D. Turnbull and N. Alldrin, "Failure prediction in hardware systems," University of California, San Diego, Tech. Rep., 2003. [Online]. Available: http://cseweb.ucsd.edu/~dturnbul/Papers/ServerPrediction.pdf

[27] B. Zhu, G. Wang, X. Liu, D. Hu, S. Lin, and J. Ma, "Proactive drive failure prediction for large scale storage systems," in *Mass Storage Systems and Technologies (MSST), 2013 IEEE 29th Symposium on*. IEEE, 2013, pp. 1–5. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6558427

[28] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, Apr. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?doid=1961189.1961199

[29] J. Li, X. Ji, Y. Jia, B. Zhu, G. Wang, Z. Li, and X. Liu, "Hard drive failure prediction using classification and regression trees," in *Dependable Systems and Networks (DSN), 2014 44th Annual IEEE/IFIP International Conference on*. IEEE, 2014, pp. 383–394. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6903596

[30] A. Sands and V. Tseng, "1 in 3 Laptops fail over 3 years," SquareTrade, Inc., Tech. Rep., Nov. 2009. [Online]. Available: https://www.squaretrade.com/htm/pdf/SquareTrade_laptop_reliability_1109.pdf

[31] "Consumer Reports: Extended Warranties," ConsumerReports.org, Tech. Rep., 2010. [Online]. Available: http://www.consumerreports.org/cro/magazine-archive/2010/december/electronics/best-electronics/extended-warranties/index.htm

[32] "Global Efficient Call Center Salary," 2016. [Online]. Available: http://www.indeed.com/salary/Global-Efficient-Call-Center.html

[33] J. Gaw, "Pivot Table: Worldwide Consumer Market Model, 20102019," Aug. 2015. [Online]. Available: http://www.idc.com.libproxy.mit.edu/getdoc.jsp?containerId=258735

[34] K. Tofel, "Only 5 percent of IBM's Mac and iOS users call support, compared to 40 percent of Windows users," Oct. 2015. [Online]. Available: http://www.zdnet.com/article/only-5-of-ibms-mac-and-ios-users-call-support-compared-to-40-of-windows-users/

[35] "Desktop operating system market share 2012-2015." [Online]. Available: http://www.statista.com/statistics/218089/global-market-share-of-windows-7/

[36] Y. Ali, "A Fuzzy Model For Surface Grinding," Ph.D. dissertation, The University of Sidney, Sidney, Mar. 2007. [Online]. Available: http://yasserali.org/?cat=30