Aalto University
School of Science
Degree Programme in Computer Science and Engineering

Markus M. Virtanen

Predictive analytics for improved problem management

Master's Thesis
Espoo, March 28, 2014

Supervisor: Associate Professor Keijo Heljanko
Advisor: Esa Manninen M.Sc. (Tech.), M.Sc. (Econ.)

Aalto University
School of Science
Degree Programme in Computer Science and Engineering

ABSTRACT OF
MASTER'S THESIS

Author: Markus M. Virtanen

Title:
Predictive analytics for improved problem management

Date: March 28, 2014 **Pages:** 50

Major: Software Technology **Code:** T-220

Supervisor: Associate Professor Keijo Heljanko

Advisor: Esa Manninen M.Sc. (Tech.), M.Sc. (Econ.)

This Thesis studies the possible data analysis methods to improve the problem management service at Elisa corporation. The primary objective is to find and evaluate appropriate data analysis methods to understand the underlying problem causes and to develop predictive models from hypotheses and data, so that the problems could be prevented or minimized before the customer feels she or he has to contact the customer service.

Apache Hadoop framework and R programming language have been integrated to make a linearly scalable analytics process possible. Customer data is first imported to research computer so that there is no need to execute analytical queries on the production databases.

The analytics phase is divided into two parts both helping with data-based decision making. At first, exploratory data analysis is used to find valuable information from the historical data with the help of textual data summaries and visualizations. After that, when there is cleaned data and a good understanding has been obtained of its behaviour, predictive analytics is included to make forecasts about the future behaviour.

Out of all the evaluated machine learning algorithms for the given task, the Random Forest algorithm had the best performance. The final tuned model predicted the right outcome for approximately 77 % of problem tickets minimizing the need to send a repair technician if a customer's Internet connection works well.

Keywords: data analysis, predictive analytics, geostatistics, big data, machine learning, random forest, Hadoop, Hive, R

Language: English

Aalto-yliopisto
Perustieteiden korkeakoulu
Tietotekniikan koulutusohjelma

DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Markus M. Virtanen		
Työn nimi:	Ongelmanhallinnan parantaminen ennustavalla analytiikalla		
Päiväys:	28. maaliskuuta 2014	**Sivumäärä:**	50
Pääaine:	Ohjelmistotekniikka	**Koodi:**	T-220
Valvoja:	Professori Keijo Heljanko		
Ohjaaja:	Diplomi-insinööri, KTM Esa Manninen		

Tämä diplomityö tutkii mahdollisia data-analyysimenetelmiä parantaakseen Elisan vianhallintapalvelua. Ensisijainen tarkoitus on löytää ja evaluoida sopivia menetelmiä, jotka antavat ymmärrystä vikoihin, ja luoda ennustemalleja hypoteesien ja datan perusteella, joilla ongelmat saataisiin estettyä tai minimoitua ennen kuin asiakkaan tarvitsee ottaa yhteyttä asiakaspalveluun.

Apachen Hadoop ohjelmisto ja R-ohjelmointikieli integroitiin tehden lineaarisesti skaalautuva analyysiprosessi mahdolliseksi. Asiakkaan data haettiin ensin tutkimustietokoneelle, jotta analyyttiset haut eivät kuormita tuotantotietokantoja.

Analyysivaihe on jaettu kahteen osaan, jotka molemmat auttavat dataan perustuvaa päätöksentekoa. Ensin tutkivia analyysimenetelmiä käytetään löytämään arvokasta informaatiota historiallisesta datasta hyödyntäen tekstipohjaisia yhteenvetoja ja visuaalisia menetelmiä. Seuraavaksi, kun data on siivottua ja siitä on hyvä käsitys, prediktiivistä analyytiikkaa käytetään luomaan ennusteita tulevasta.

Kaikista tarkastelluista koneoppimismalleista Random Forest algoritmilla oli paras suorituskyky annettuun ennustetehtävään. Lopullinen optimoitu malli ennusti oikein 77 % kaikista ongelmatiketeistä vähentäen tarvetta lähettää korjaaja kentälle asiakkaan Internet-yhteyden toimiessa hyvin.

Asiasanat:	data-analyysi, ennakoiva analytiikka, geostatistiikka, big data, koneoppiminen, random forest, Hadoop, Hive, R
Kieli:	Englanti

Acknowledgements

I would like to thank Associate Professor Keijo Heljanko for his ideas and guidance in the Thesis project.

From Elisa's problem management, I would like to thank Esa Manninen, Petri Kononow, Aleksi Heikkilä and Jari Hyppänen for their domain knowledge on network related data, and Jukka Otajärvi for his help with geospatial data.

From Elisa Viihde team, I would like to thank brothers Timo Hakkarainen and Leo Hakkarainen for helping with the Set-Top-Box related data and issues.

I also would like to thank Antti Kallonen for analytical help in R, Ismo Hannula for his help in regressions and statistics, Jari Mieho for discussions about computer hardware, Andon Nikolov for network related analysis ideas, and Paul Choo for his knowledge of time-series analyses in financial problems.

Lastly, I would like to thank my family for their support during my studies and uncle Jukka for starting my interest towards computing and Pascal 25 years ago.

Espoo, March 28, 2014

Markus M. Virtanen

Abbreviations and Acronyms

ADSL	Asymmetric Digital Subscriber Line; technology for data transmission over copper telephone lines.
CSV	Plain text file with Comma-Separated Values.
Convex Hull	Set of points in Euclidean space that creates the smallest convex.
DSLAM	Digital Subscriber Line Access Multiplexer; network device for connecting several DSL interfaces to network backbone and separate voice and data traffic.
DVB-C	Digital Video Broadcasting - Cable. European-based consortium standard for digital television over cable transmission.
DVB-T	Digital Video Broadcasting - Terrestrial. European-based consortium standard for digital terrestrial television transmission.
ECDF	Empirical Cumulative Distribution Function used with the empirical measure of a sample.
EDA	Exploratory Data Analysis. Approach to summarize data sets main characteristics.
Elisa Viihde	Set-Top-Box that adds IPTV services to regular TV transmissions.
ETRS-TM35FIN	Finnish extension for Universal Transverse Mercator coordinate system zone 35.
HDFS	Hadoop Distributed File System.
HTTP	HyperText Transfer Protocol.
IP	Internet Protocol, principal communications protocol in internet protocol suite.
JDBC	Java DataBase Connectivity; interface defining how a database can be accesses.
KPI	Key Performance Indicator; performance measurement to evaluate organization's success.

MCAST	Multicast, delivery of information to a group of destination computers using a single transmission from the source.
MPEG-2	Standard format for coding of video and audio data.
NPS	Net Promoter Score; customer loyalty metric.
RAM	Random-access memory; form of computer data storage that can be accessed directly in any random order.
ROC Curve	Receiver Operating Characteristic Curve; a plot for visualizing efficiency of machine learning models.
SD	Standard Deviation; average dispersion of a set of data.
STB	Set-Top-Box.
SVM	Support Vector Machine; supervised machine learning model used for classification and regression analysis.
TCP	Transmission Control Protocol; provides reliable, ordered and error-checked delivery of stream of octets.
UDP	User Datagram Protocol; simpler mechanism for IP-messaging without error-checking and correction.
VOD	Video On Demand; system that allows a user to watch video content on demand.
YKJ	Yhtenäiskoordinaatisto, Finnish uniform coordinate system.

Contents

Chapter 1

Introduction

When a corporate problem management unit starts to identify customer's service quality, they usually have several data sources to inspect instead of just a single system that actually hosts a customer service. For a telecommunication company this could include lots of data from technical systems such as network devices, subscription provisioning systems and customer service system if the customer has already contacted the company about the level of service.

As volume, velocity and variety of data continuously increase, the corporate needs to think of what parts of systems and data are considered important and how close to real-time there should be reports about their service levels. As an end-result, there should be a design for data analysis architecture that can store what is required and allows efficient analysis for proactive methods such as total incident prevention or impact minimization when total prevention is not possible.

Data analysis is a process whose main objectives are to find valuable information, suggest conclusions and support decision making. Predictive analytics is a part of data analysis where something is forecasted.

The goal of this thesis is to demonstrate different data analysis methods from both exploratory analysis and predictive analysis for problem management purposes. Main focus for the exploratory part is given to a Set-Top-Box device. The predictive analysis is performed to forecast unnecessary technical repair person visits for DSL-related problem scenarios.

The practical part for this Thesis was done in the fall of 2013 and the beginning of 2014 after it was requested by the client. The client for this Thesis is the Problem Management unit of Finnish telecommunication company Elisa.

1.1 Problem Statement

Elisa is looking for methods that can increase proactive problem management. That is, for example, problems in network, devices and service processes, which can be monitored and fixed before the customers have to contact the customer service about a problem.

To understand the current situation, we have to have an analytics platform, and analytical tools to separate normal and abnormal activity in the statistical characteristics of the service data. With a good understanding, we can create prediction models to forecast the future thus making also the proactive methods possible.

This Thesis describes and implements a scalable data warehouse platform with analytical tools and methods that can be used for automated analytics.

1.2 Structure of the Thesis

This Thesis first introduces the basics about data analytics and company problem management. Then the client's working enviroment and the available data is described. After that, possible data analysis methods for the given data are shortly introduced and split to two groups; exploratory data analysis and predictive modeling. In the implementation chapter, the data analytics environment is first designed and implemented to a research computer. Then the given data is being analysed and the analysis results are gathered. In the end of chapter, a predictive problem ticket model is created. Finally, the evaluation chapter describes some further notes about the implemented solutions.

Chapter 2

Background

2.1 Data Analysis

Data analysis is a process whose main objectives are to find valuable information, suggest conclusions and support the data-based decision making. Data analysts work close to business issues and an important criteria to keep in mind when doing data analysis is to think what benefits [10] does the analysis bring to the company.

By using data analysis, we can look for interesting patterns and behaviours from even bigger amount of data and still concentrate on the individual level of a single customer. For example, marketing could use such information for one-to-one marketing and tailor campaigns directly to an individual. Such personalization is already part of several web sites and consumers have started to understand that data is being collected on them. On the other hand, individual level data analysis brings privacy issues and increases demand for Privacy-Preserving Data Mining, but it also makes the customers to expect more intelligent interactions and experiences that includes their uniqueness instead of general options.

The analysis process starts by understanding the phenomena from historical data. We try to create models that present the data. The better a model fits the data, the better and more profitable are the future predictions we can achieve based on it. A famous prediction competition was the *Netflix Prize* where user ratings for films were best predicted by the *BellKor's Pragmatic Chaos* team's solution [19].

2.2 Big and Fast Data

Big data is still relatively new topic in IT, for both developers and industry, thus still lacking standardization efforts. More sensors are being used when electronics get smaller and cheaper, more online user actions are being stored and utilized to better understand customer experience, and the worldwide availability of high-speed Internet connections brings more "cloud" services [7]. Volume, velocity and variety of data is continuously increasing. This brings problems to analytics such as can we capture and store the data efficiently and securely. On the other hand, can it be cleaned, enriched and processed in acceptable time without sampling, so that those millions of records in the database can be transformed into knowledge without a need to move into shorter observation periods. Finally, are the results accessible for searching, integration and visualization - how to, for example, make a scatter plot of 100 million measurements and show it in an image with resolution of 640x480 pixels.

More organizations have started to open their databases to the public so that companies and individuals can enrich their own measurements by, for example, demographic, geographic and weather data. In Finland, open data resources can be received from sources such as FMI Finnish Meteorological Institute, NLS National Land Survey of Finland and Statistics Finland.

Making your code execute few milliseconds faster means great advantage when there are millions of records to be processed in a big data environment. For the scope of this Thesis, we define big data to be at least in order of terabytes in size of the original dataset. Also smaller datasets can be considered as big data according to the biggest object created during the analysis process, for example, when a distance matrix is calculated.

2.3 Problem Management

Problem Management is the part of company that identifies problems in products and services, and prevents resulting incidents from occuring.

Problem management involves: root-cause analysis to determine and resolve the cause of incidents, proactive activities to detect and prevent future problems/incidents and a Known Error sub-process, to allow quicker diagnosis and resolution if further incidents do occur [11, p. 14].

To succeed in this, problem management has to have an access to all problem related data. Usually this includes systems such as:

- technical systems and services themselves where the problem occurs,

MAC	START	END	MEDIUM	EKEY	EVAL
00a000efe200	2013-09-12 16:47:13	2013-09-12 16:51:13	DVBC	Errored seconds	5.000

Table 2.1: Example of STB device error log event - during a four minute monitoring interval on cable TV, 5 seconds were considered as errors.

- provisioning or delivery process systems handling the service towards customers after the payment, and
- customer service systems when customer has already contacted about a service.

For problem management purposes, data analysis includes for example fault isolation and searching for correlations what could be the root-cause for a specific problem. It can be used for traffic planning to verify that network devices are getting enough throughput for a service to work. Another option is to use it for predictive hardware maintenance by estimating how long is the service life-cycle before an error occurs. On the other hand, it can be used to look for chronic circuit problems to identify if there is something wrong, such as bad soldering, with factory batch of devices.

The main problem management goal for this thesis is to understand methods how Elisa's STB device's key performance indicator (KPI) can be increased by decreasing the device unavailability-%.

2.4 Elisa Viihde

Elisa Viihde has several set-top-box (STB) models for video entertainment and this Thesis concentrates on one of the models. The STB is used with four different mediums; Digital Video Broadcast - Cable (DVB-C), Digital Video Broadcast - Terrestrial (DVB-T), Multicast and Video On-Demand (VOD), out of which Multicast and VOD work on top of an IP-network.

STB device's reporting mechanism shows the quality of these mediums by seconds - how many seconds was a medium watched and how many seconds of that were considered as error seconds. Table 2.1 demonstrates a device error log event.

Multicast is used to deliver tv channel data to a group of destination addresses simultaneously within a single transmission. It uses the User Datagram Protocol (UDP) and thus the transmission is not reliable and messages may be lost or they might get delivered out of order. On the other hand, no extra

bandwidth is consumed to ensure reliability, in contrast to the Transmission Control Procotol (TCP).

VOD service includes both the movies a customer can rent, and personal video recordings she or he has recorded to watch them later. Transmission difference here is that the movies are transmitted using UDP, but personal recordings are sent on TCP/HTTP utilizing more customer's Internet connection uplink.

STB is currently most often sold with ITU G.992.5 ADSL2+ standard connection, limiting the connection speeds to 24 Mbps for downlink and 1 Mbps for uplink.

2.5 Problem Tickets

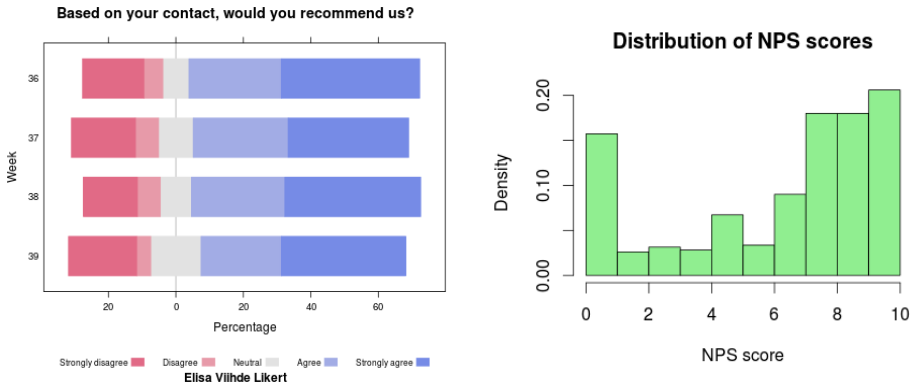
When a customer is having a problem with the service, they call to Elisa customer service that tries to fix some regular problems, such as checking whether the cables are connected the right way and the device actually gets electricity. If that does not help, the incident is still forwarded to technical helpdesk and finally to field representatives, who actually visit a customer to fix the problem. It is possible that a customer visit is performed when there is a more general network problem or a provisioning system delay thus causing an unnecessary chargeable field visit.

In this Thesis we will generate and evaluate a prediction model, which estimates probability of an unnecessary field visit thus making it possible to save money in service costs.

2.6 Net Promoter Score

Elisa uses a Net Promoter Score (NPS), which is a customer loyalty metric. When a customer is contacted she is asked how well, based on this contact, would she recommend Elisa to others, on scale 0 to 10. Elisa interpretes the results by removing grades 7 and 8 as they are considered as passive people. The model categorizes respondents giving a 9 or 10 as real promoters. This demands personnel to push harder as the NPS mean value is getting lower after the removal process.

As an example, I transformed one batch of NPS results into a Likert psychometric scale, which is used in research involving questionnaires. This was done by transforming grades 0, 1 and 2 to indicate *"Strongly disagree"*, and next four categories correspond to 2 subsequent grades each.



(a) NPS results on Likert scale. (b) Histogram of NPS grades.

Figure 2.1: Visualized NPS statistics.

In the Likert Figure 2.1a, blocks for weeks 36 and 38 are more right indicating a better overall result. Elisa model would not have the *"Agree"* group included.

Distribution of NPS grades is visualized in Figure 2.1b and it indicates that customers are usually giving either grade 0, 8, 9 or 10, with a median of 8.

I did not find any relation between the survey results and customer's actions in the real world, and I did not receive any such earlier measurement model so this Thesis concentrates more on analysis of the technical side. In my opinion though, if there is only a single question asked from me, as a customer, I generally give the opinion as an overall feeling about target company, and thus, the result might have nothing to do with the latest connection about a particular service.

2.7 Tools for Analytics

2.7.1 R Language

R [1] is an open source programming language for statistical computing and in the end of 2013, it had nearly three million users. R, created by Ross Ihaka and Robert Gentleman in 1993, is an implementation of the S statistical programming language. The primary reasons for taking R to the Thesis technology stack is that it already includes lots of data analysis functions and thus it is fast to move from cognitive thinking process into programmed implementation, and on the other hand, it provides easy data exploration properties to display the meaning in the data.

R holds all objects in memory [14] and the amount of available RAM is usually the bottleneck for data analysis. On 64-bit architectures though, the executables will have a system-specific limit, such as 128 TeraBytes with Linux on x86-64 processors. R does not have similar software lifecycle tools as Java, for example, making larger R-based applications harder to maintain.

2.7.2 Apache Hadoop

Apache Hadoop [5] is an open-source software framework for storage and large scale processing of data-sets. Unlike traditional single-thread unix tools such as sed and awk, hadoop provides parallel processing for data analytics on top of its Hadoop Distributed File System (HDFS). Data retrieving queries have to be written as MapReduce jobs, but as they are tested on a local machine, they can as well be distributed to a cluster of machines without code changes. Performance can be improved via code and configuration tuning as well as by increasing the number of cluster nodes. Hadoop is designed for sequential I/O and not for small files, random access, high transaction rates nor frequent updates, such as relational databases are.

Hadoop's design can achieve high data throughput, but it is not designed for low latencies that interactive data analysis requires. Still, there are solutions such as Google Dremel [20] and Apache Drill [2] addressing the latency requirements with the help of online algorithms [6, p. 14].

On cluster environment, individual Hadoop nodes do not share hardware resources meaning the machines are independent of each other, connected only through the network. Hadoop framework detects failures, such as hard drive failures and node-losses, and re-executes jobs on another node. The basic functionality of MapReduce is that when a query job is executed, the input data is split to smaller chunks and each chunk is trasmitted to different Hadoop node for processing. When these basic MapReduce inputs and outputs are chained sequentially for more complex queries, we talk about MapReduce pipelines.

Apache Sqoop [4] is a tool designed for transferring data between Apache Hadoop/Hive and structured datastores, such as relational databases.

2.7.3 Apache Hive

Apache Hive [3] is a data warehouse infrastructure built on top of Hadoop for providing data analysis without the need of writing complex MapReduce jobs. With the help of Hive, we can utilize SQL-like HiveQL language for data query and summarization over HDFS stored data. HiveQL uses syntax similar to MySQL and translates the queries into MapReduce jobs. The

relation between Hadoop and Hive within our implementation is illustrated in Figure 5.1. Basic usage examples and general Hive architecture is well described in paper [22].

Chapter 3

Environment

3.1 Production Databases

The original data sources for this Thesis include purely relational production databases. Being a production database means that the analytics developer should not make changes to the original data and especially that those systems might be under heavy load thus making it harder to perform comlex SQL-queries for analytics. For these reasons and research nature of project, bulk data was first retrieved to the analytics PC using simple SQL-SELECT statements and the Apache Sqoop tool.

The total amount of data to be processed is 30 GB. It contains data from Elisa Viihde STB devices, electrical measurements from DSL lines, DSLAM syslog event data, spatial location data, problem ticket and field troubleshooting data, and customer feedback data.

The error reporting system for STB devices is problematic for data analysis, because it does not provide any database dumps. Instead, one has to collect a separate CSV file for each device one by one. This sequential backup process also runs for days causing difference in error log dates.

Elisa provided YKJ coordinates for its customers. R spatial packages do not support YKJ data so well and thus they have to be transformed into WGS84 or UTM-35 coordinate system first.

Chapter 4

Methods

Within this thesis, I split the data analysis methods to exploratory part and predictive part. Exploratory Data Analysis (EDA) means ways to summarize and visualize the main characteristics of data. Predictive analytics is an approach for analysing historical data to make predictions about future.

The whole data analysis project is an iterative process, where much of time is consumed in data cleaning and transforming it to an appropriate format for analysis. In modeling phase, the analytics developer often realizes need for further cleaning and transforming.

4.1 Exploratory Data Analysis

The goal of EDA phase it to discover new knowledge. We can use tools such as boxplots, scatter plots, histograms and time-series plots to turn data into information. In the end of EDA phase, when we understand the past, we can continue to predictive analytics to make predictions about the future. We also go back to EDA from predictive analytics whenever we need data cleaning, dimensionality reduction or new predictors so we consider the whole process as an iterative one [24, p. 301].

4.2 Spatial Analysis

Elisa has network infrastructure around the country and each device installation is influenced by unique features of the network. Spatial analysis is one solution to understand which customers and which parts of the network topology influence most the STB device error rates. Spatial analysis is a subfield in statistics, where each observation is bound to a specific location. Since we know the device locations with their error rates, we will utilize

Kriging [16, p. 165] to create a geostatistical probability model to estimate the geographical situation also for the unknown locations. The Kriging approach provides also prediction error estimates that are discussed further in Section 6.1.

4.3 Predictive Analytics

The predictive task within thesis was to predict which customer problem tickets will need no actual work to be performed. We will use the EDA results to achieve understanding about the current data model and then, create a prediction model. The prediction model is based on supervised learning, which means that our application observes some example input-output pairs and learns a function that maps from input to output [21, p. 695]. The model has to be able to generalize [18, p. 2] what it has learned from historical data, meaning it has to be able to make reasonable decisions for inputs that were not encountered during the model training.

British mathematician George E. P. Box[1] stated that *essentially, all models are wrong, but some are useful*. This means that even if our prediction model is well-trained, it most probably can not forecast everything right but instead, we are searching for a model that performs efficiently enough.

4.3.1 Machine Learning

Machine Learning (ML) provides tools to transform data into knowledge. Machine learning can be split to classification and regressions algorithms, where classification algorithms try to identify group memberships for classes and regression algorithms estimate continuous value responses. Several ML-algorithms provide dual usage, and thus it can be used for both types of problems.

From a programmer point-of-view, the ML-algorithms can be considered as black boxes - each take similarly input data and provide output data by utilizing simple or more advanced mathematics on the background, and they leave the user mainly the task of providing parameters to tune the algorithmic efficiency for a specific dataset.

For the problem ticket predictions, we use the classification method which is supported by algorithms such as Logistic Regression, Decision Tree, Support Vector Machine, Artificial Neural Network, Adaptive Boosting and Random Forest.

[1] http://en.wikiquote.org/wiki/George_E._P._Box

Logistic Regression belongs to generalized linear models and it fits a regular regression curve to predict a binary outcome based on one or more features in the data [13, p. 385]. The binary parameter describing the possible outcomes of the model, is generalized to 0-1 range by using a logistic link function [17]. Logistic Regression can provide a fairly good reference model when the predictor variables are continuous.

Decision Tree has a similar structure as flowcharts [23, p. 352]. Each decision tree represents a certain aspect of the whole dataset. They are relatively fast to construct and if the tree size is small, the tree rules are interpretable without statistical skills, since the knowledge is presented as logical structures. Certain industries could require a possibility to explain how the forecasted decision was made. For example, if a loan application is rejected, the customer could be interested in the reasons how he or she was not accepted.

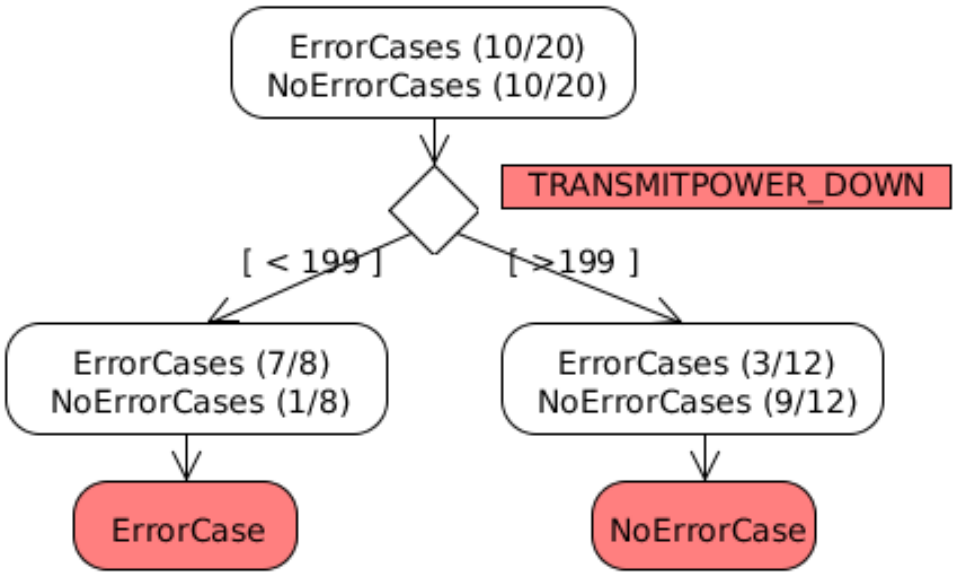


Figure 4.1: An example decision tree that splits Error and NoError cases based on TRANSMITPOWER_DOWN value. If TRANSMITPOWER_DOWN is less than 199, the classification makes an assumption about Error outcome.

In the Figure 4.1 we have only two terminal nodes so the tree assigns every record to one of the two leaves. From 20 transmitpower records, the tree assigns more error outcomes when transmitpower is less than 199. There

are still a few outcomes that will get predicted incorrectly increasing the uncertainty of the decision tree model. Uncertainty brings the problem when our prediction model might get great structural changes with even a small variation in the training data. If we would combine several decision trees with their somewhat average outcomes, the model uncertainty should get reduced. Such combined models are called an ensemble of decision trees or *Random Forests* [15, p. 320]. Such combinations can bring better prediction performance when the downside is loosing the interpretations and explanations of the model logic.

Adaptive Boosting belongs to ensemble learning methods, like Random Forests, and it utilizes collections of statistical classifiers which are more accurate than a single classifier [12]. The key feature in boosting is that instead of equal averaging of results from each model, a weighted average is calculated. The weights are based on each model's performance. Adaptive Boosting looks for records that seem to be hard to predict and gives less attention to records that were already correctly forecasted.

Support Vector Machine (SVM) and Artificial Neural Network (ANN) belong to more mathematical algorithms. Greater complexity might mean longer execution times but also more possibilities for model tuning. The Artificial Neural Network is a parallel distributed processor made up of simple processing units, which stores experimental knowledge and makes it available for use [18, p. 2]. The Support Vector Machine [15, p. 338] uses multidimensional surfaces and a maximal margin classifier to find the relationship between predictor variables and outcomes.

We will evaluate the different classification methods by using a Receiver Operating Characteristic (ROC) curve and verifying how many actual observations were predicted correctly. After the initial model, we will try to calibrate the model by tuning possible algorithm parameters.

Chapter 5

Implementation

5.1 Architecture for Analysis

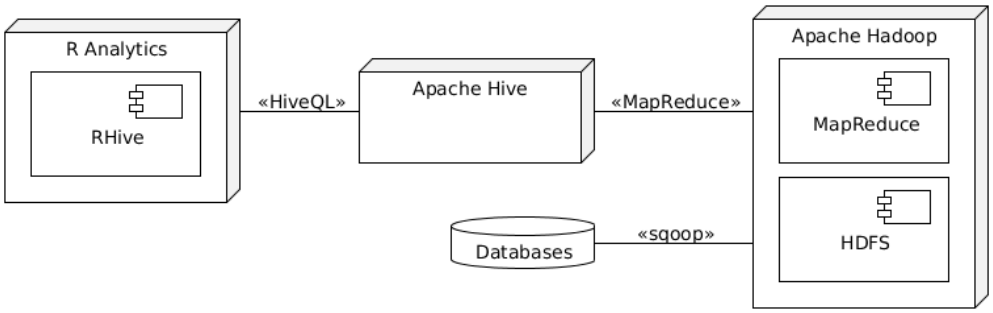


Figure 5.1: Deployment diagram describing the implemented data analysis architecture.

Whole analysis was performed on a single computer with Intel Core i5-2520M quad-core processor, 8 GB of RAM, and 240 GB SSD drive. The software stack included Ubuntu 12.04 LTS, R 3.0.2 a.k.a Frisbee Sailing, Apache Hadoop 1.0.2 and Apache Hive 0.12. The software stack with its connections are illustrated in Figure 5.1. Most of Elisa's data was first downloaded to Hadoop's file system using Apache Sqoop. Apache Hive was started in server-mode it provided a connection between RHive package and Hadoop. In the end, we were able to get appropriate data to R as offline batch jobs [8] by using RHive and MySQL-like syntax.

To speed up smaller analysis, some tiny datasets, such as problem tickets, were connected straight to R analytics by using traditional CSV-files instead of getting them through Hive and Hadoop.

After initial installations, Hadoop was benchmarked to see it is set up correctly. TestDFSIO tests the I/O performance of the HDFS by using a MapReduce job to read or write files in parallel [25, p. 331]. In our benchmark, we wrote 10 files, each size of 1 GB, and that corresponds to 10 MapReduce jobs and 10 GB of data.

```
----- TestDFSIO ----- : write
           Date & time: Sat Oct 26 20:06:38 EEST 2013
       Number of files: 10
Total MBytes processed: 10000
     Throughput mb/sec: 127.02445220704986
Average IO rate mb/sec: 127.88526916503906
 IO rate std deviation: 10.377000979835607
    Test exec time sec: 82.675

----- TestDFSIO ----- : read
           Date & time: Sat Oct 26 20:09:20 EEST 2013
       Number of files: 10
Total MBytes processed: 10000
     Throughput mb/sec: 202.8479857195018
Average IO rate mb/sec: 202.8806610107422
 IO rate std deviation: 2.5799491941344113
    Test exec time sec: 70.591
```

According to benchmark, our Hadoop implementation can write nearly 130 MB and read more than 200 MB a second. This is fairly good speed with the utilized hardware.

5.2 Exploratory Data Analysis

We will start by trying to understand how the STB device errors are distributed. We start by ordering the individual device errors from max to min and then use Empirical Distribution Function (ECDF) to plot the cumulative distribution function.

Customer	Median	Min	Max
1	86642.5	47175	86675
2	86622.0	35127	86675

Table 5.1: Daily error seconds of the worst STB customers. Median error seconds are greater than there are seconds in day indicating problems with the reporting system.

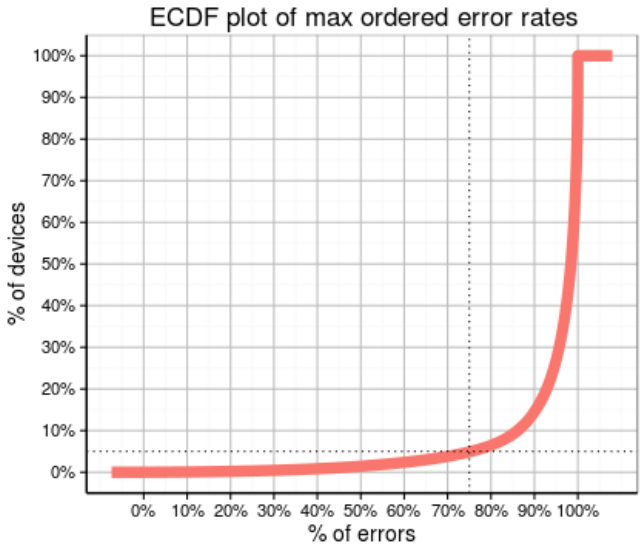


Figure 5.2: ECDF plot of Elisa Viihde device errors - 5 % of worst devices contribute 75 % of all errors.

The main message from the ECDF plot in Figure 5.2 is that only 5 % of top errornous devices are responsible for 75 % of all the errors we are dealing with. Worst devices actually report more than 86 400 error seconds a day (60 sec * 60 min * 24 hours = 86400 seconds/day) indicating a problem within the error reporting system.

In MPEG-2 standard, video stream compression can utilize delta frames. Instead of sending one fully specified image in each frame, there is a single image frame and delta frames specifying only changes in the image from previous frame. This saves space for example when there is some stationary background and one person moving in the video. It was pointed out that STB device reports an error when a user seeks a video stream and presses play button while on delta frame. The STB considers this as missing image data error.

Using a 24h time-series plot and both usage and error seconds combined, we can see if there is a time of the day when more errors occur. Such scenario could mean for example a rush hour when a network element gets too high utilization and thus increasing the response times.

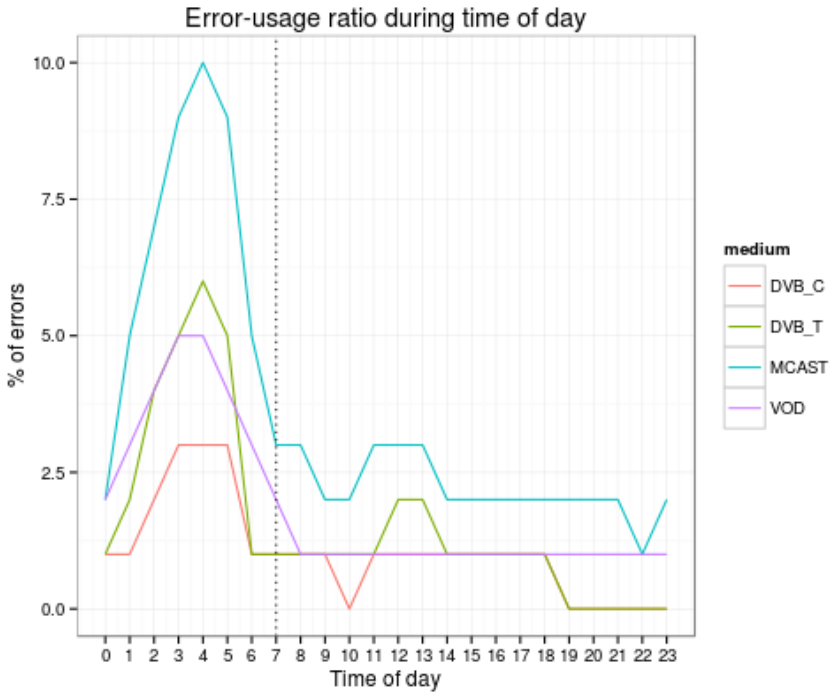


Figure 5.3: 24-hour time series plot indicating greater error ratios during the night time.

In Figure 5.3 we can see a huge increase in error amounts during night time 00-07. The reason for this phenomenom is that there is not much usage during those hours, but the broken devices mentioned earlier, with even 86400 error seconds a day, continue reporting and therefore making the increase in errors vs usage plot.

Especially the multicast medium is getting higher error counts during night. This is partly because some tv channels stop broadcasting empty signal when there is no tv program available, making STB report errors.

Next we aggregate the device error seconds to daily amounts and use a histogram to see how individual devices experience errors during a normal day.

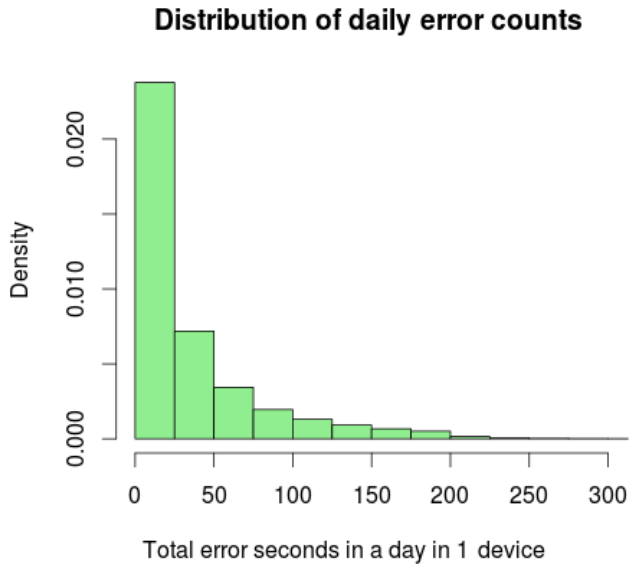


Figure 5.4: Histogram of daily device errors

In Figure 5.4 we can see that majority of devices experience a total of 0-25 error seconds each day and the group of 25-50 error seconds is already much smaller. There are hardly devices who experience more than 200 error seconds a day. For a general picture how the Viihde devices work, we concentrate on devices with less than or equal to 200 daily error seconds.

Next we try to understand better how these error seconds are made of different mediums using a boxplot visualization.

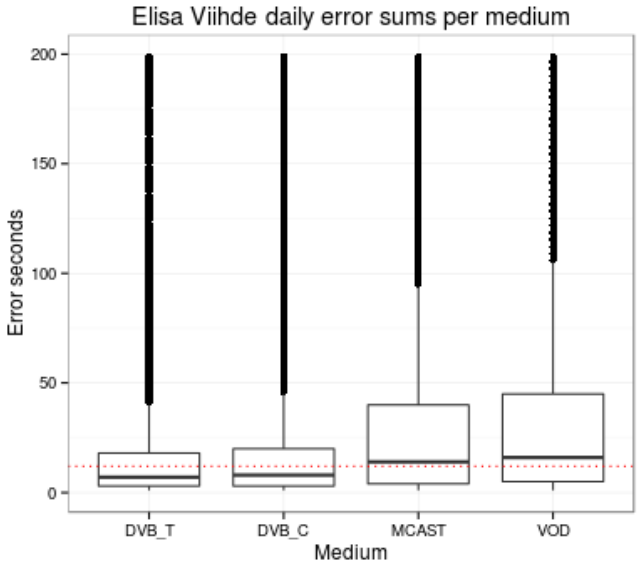


Figure 5.5: An ordered boxplot of daily device errors per medium. Red dashed line represent a generic daily error median of 12 seconds per device.

The boxplot in Figure 5.5 shows that the terrestrial (DVB_T) and cable (DVB_C) television mediums have less daily error seconds than the IP-protocol mediums MCAST and VOD.

5.2.1 ADSL Modem Firmware Update

Some of the Elisa ADSL modems are remotely managed and updated. During the Thesis process, Elisa tested one modem firmware update that brought Seamless Rate Adaptation (SRA) feature to ADSL2+ devices. Modems are affected by interferences such as cross talk from adjacent lines and temperature changes. SRA helps the modem to accommodate changes in data transfer rates with less or no need for dropping and renegotiating the whole connection. Network sees these drops as DSL up/down events and problematic connections can be assigned to safe mode profiles with a weakened transfer rate. With the help of SRA, these safe mode profiles are not needed, and connection transfer rates can automatically increase when link quality is improved.

Nearly 100 ADSL modems received the SRA update and we recorded the device behaviour data for two weeks before and after the update. Special interest was given to network parameters DslDownCount and DslUpCount indicating a line drop and connection renegotiation. The overall picture for

those values is shown in Figure 5.6.

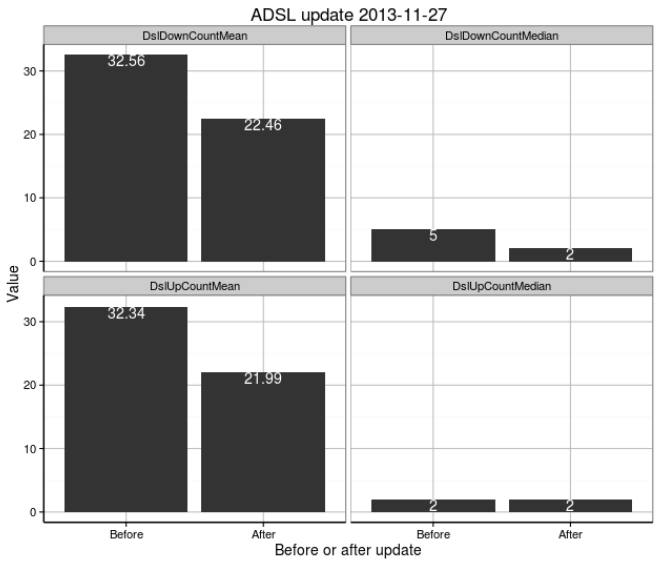


Figure 5.6: Overall picture of SRA firmware update; both median and mean values have decreased since the update indicating more stable connections with fewer line drops.

We can see that the total daily drops are decreased by approximately 30 % in its mean value and median is also lowered from 5 to 2 drops in DslDownCount. The mismatch between different amount of up and down counts are explained by features of the line monitoring software. Generally, when a line goes down, it also eventually gets up.

The dataset was further split to three groups by ADSL profiles that are normal connections, Safe Mode 1 and Safe Mode 2. We realized that there was an interesting situation at Safe Mode Group 1, whose connection speeds are lowered to provide better line quality. This situation is shown in Table 5.2. We can see that the mean values and standard deviation of connection drops have increased by some 50 % after the firmware update and at the same time the median drops have decreased. This indicates that there can be a couple of outliers who started to behave badly after the update.

It was soon noted that there was one modem, that did not like the update and increased the connection drop amounts. Situation after excluding the single device from dataset is seen in Table 5.3. Now, also the mean value has decreased and there is only a minor increase in standard deviation. Information about the analysis and the single misbehaving device was sent to problem management for further investigations.

BEFORE	Mean	SD	Median
DOWNCOUNT	12.88	24.08	4.0
UPCOUNT	12.73	24.00	4.0
AFTER			
DOWNCOUNT	19.89	42.66	3.0
UPCOUNT	18.88	41.76	3.0

Table 5.2: For Safe Mode 1 group, connection renegotiation mean and max count increased while median decreased.

BEFORE	Mean	SD	Median
DOWNCOUNT	11.36	24.46	3.0
UPCOUNT	11.17	24.32	3.0
AFTER			
DOWNCOUNT	9.81	28.84	2.0
UPCOUNT	9.47	28.61	2.0

Table 5.3: After removing a single modem from analysis, both mean and median of connection drops improved for Safe Mode 1 group.

5.2.2 Chronic Circuit Problems

By using string distance algorithms, one can calculate the distance between two strings. Device's MAC addresses are usually programmed sequentially at the factory so MAC's with close distance represent the same factory batch.

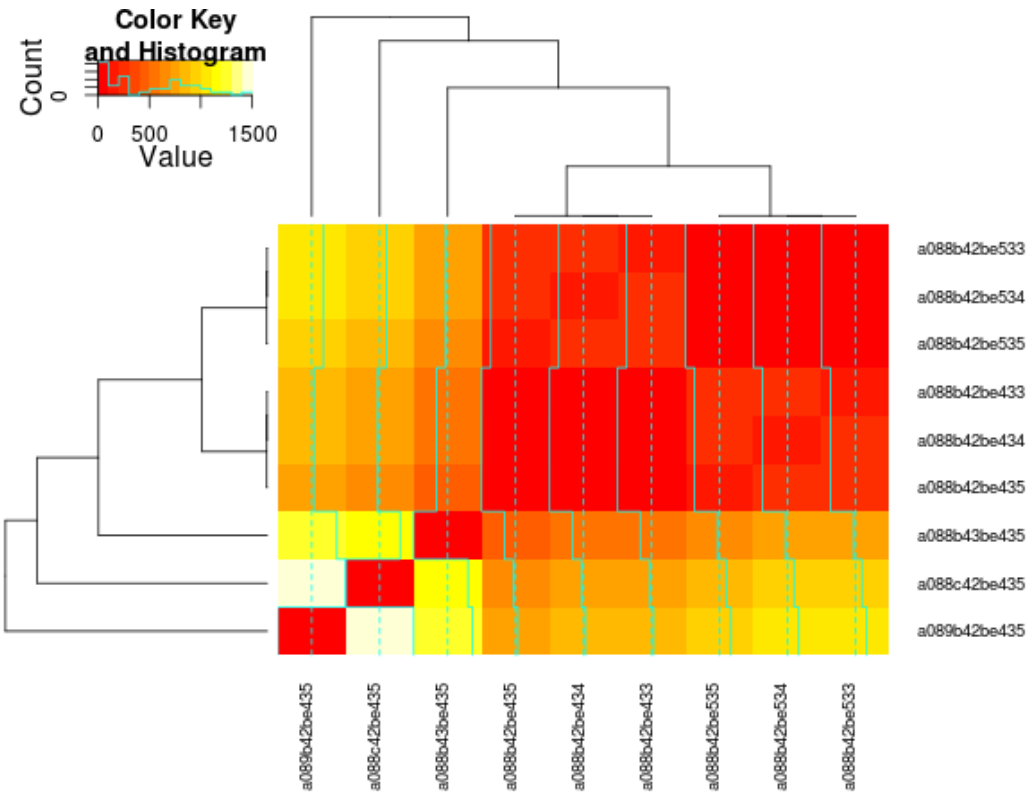


Figure 5.7: Example MAC addresses with their string distances. Red color indicates closer relationship of two strings.

In Figure 5.7 we can see two groups of three very similar MACs and three individual MACs representing different batches and therefore greater string distances. Dendogram lines were added to further reveal the string groupings. This technique was used to reveal possible batches of 100 worst STB devices. Several devices were also changed after the analysis.

5.3 Spatial Analysis

Simple kriging [16, p. 165] was used to make a heatmap of STB errors so that coordinates without error values received interpolated values. In kriging process, we compute a weighted average of the known STB errors with their coordinates, and use it to predict error rates on new coordinates in the neighbourhood. YKJ coordinates of STB customers were first transformed to ETRS-TM35FIN using EPSG Projection 2393 for YKJ and 3067 for ETRS-TM35FIN. The new coordinates were sampled randomly from whole Finland, sample size ranging from 10.000 to 20.000 points.

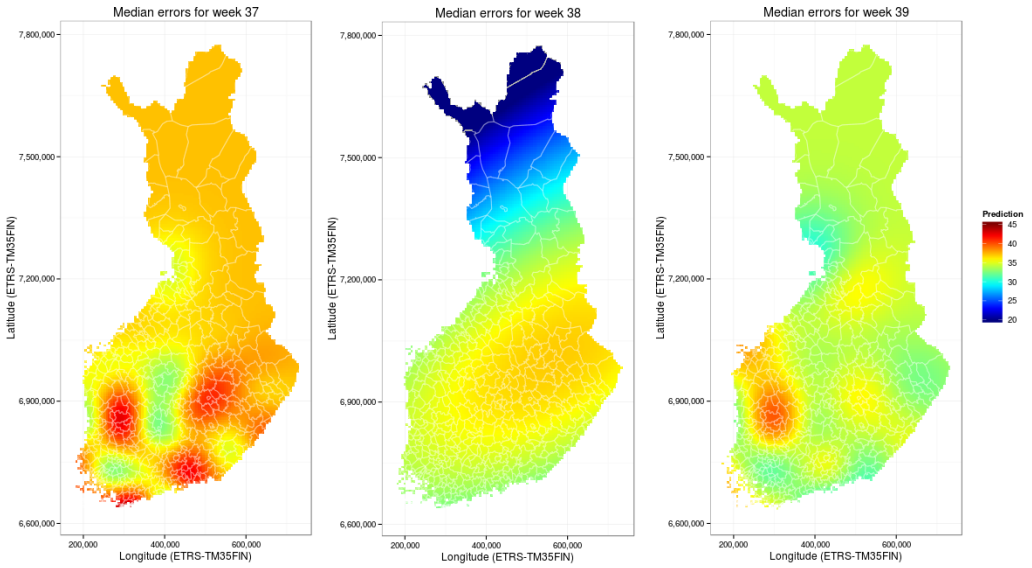


Figure 5.8: Interpolated heatmaps describing Elisa Viihde median error rates for IP-based mediums during weeks 37-39.

In Figure 5.8 we can see that the overall error rates are getting smaller as time passes. The red area in west coast seems to stay high in errors for both week 37 and 39. Generally, the areas with constant higher median error could be caused, at least partly, by longer than average ADSL cables that increase the line attenuation and decrease the connection speeds.

5.4 Predictive Analytics

In this section, we build a prediction model to estimate if a problem ticket actually requires repair work or not. For this, we used problem ticket data made up by a customer service representative and data from both copper line measurements and line renegotiation data from DSLAMs.

5.4.1 Cleaning and Combining Data

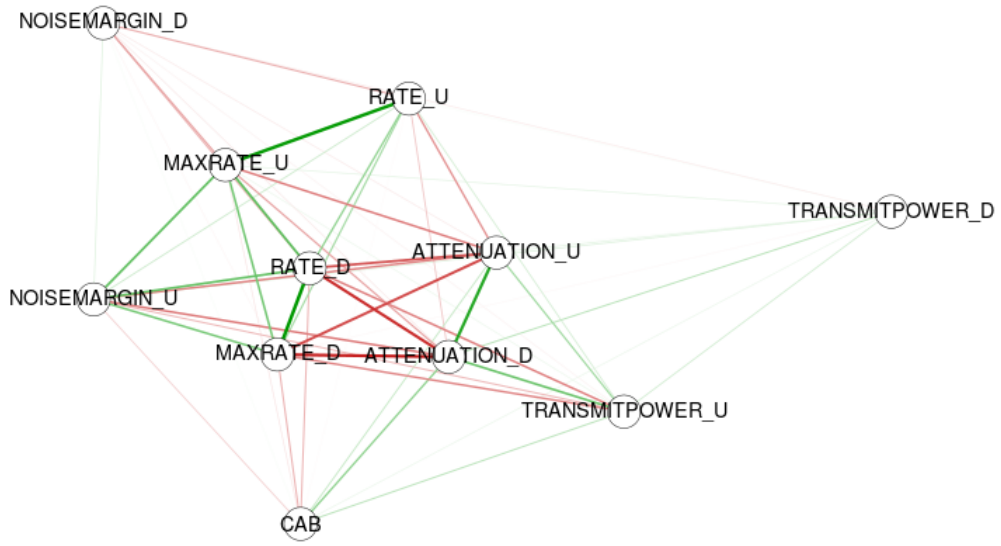


Figure 5.9: Net graph describing correlations between copper line measurements.

In Figure 5.9 we see correlations with green for positive correlation and red for negative. Positive correlation means both values increase and decrease simultaneously, for example, when attenuation down increases, also attenuation up increases. Red colored negative correlation means the opposite, for example, while attenuation down increases, data's rate down decreases and vice versa. The thicker the line is between two parameters, the stronger is the correlation.

Greater attenuation on copper line requires more transmit power and increases retransmissions of data packets. When thinking of general DSL speed of 24/1 Mbps, downwards attenuation plays a bigger role than upwards attenuation, as the frequency range for 24 Mbps data connection is both higher and wider than it is for a 1 Mbps uplink. We see the attenuation down correlates with the cable length (CAB) indicating the obvious fact that greater length of copper line increases the attenuation. To find out possible weather and time of year changes to line attenuations, an annual sample was taken for year 2013.

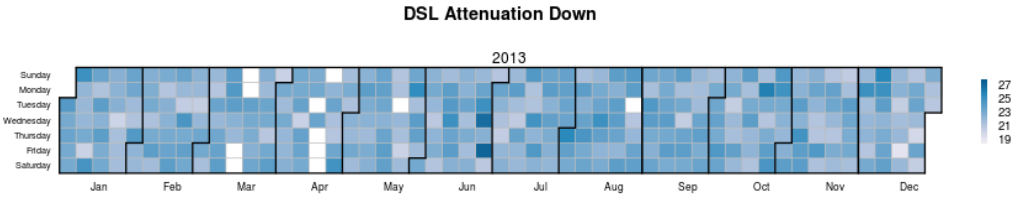


Figure 5.10: A sample with median DSL attenuations in 2013.

Figure 5.10 shows that there is hardly evidence that we should consider any greater change in line attenuations during a year. However, summer time attenuation median is slightly darker in plot and approximately 0.5-1.0 dB higher than winter time. Difference is small and could be explained by hot weather or small sample set size. Median daily attenuation changes between 19 and 27 dB during the year.

Each problem ticket has general data, such as timestamp, customer identification, full text description of the problem, who is dedicated to fix the issue and how was the problem solved in the end. Customer service agent also chooses a parameter describing the problem. Possible states for problem include "Network not functional, modem light for internet not active", "Network reconnections occuring continuously" and "Down rate speed is incorrect". In the end, when a repair representative handles the issue on the field, she chooses to field "Final result" how the issue was repaired. Possible results are for example "No problem after all" and "Fixed locally/modem firmware updated".

We cleaned and combined these datasets for a classification task so that each observation included copper line measurements, ticket data and a binary variable having value 1 when there was actually an error and 0 when there was no real error after the field visit. The final dataset included 264 observations with 13 predictor features. There were in total 110 cases of no error tickets and 154 real error tickets.

5.4.2 Choosing the Best Machine Learning Model

Different machine learning algorithms have their unique properties making them good for different use cases. We have to test different models to find out how accurate they are for our problem with its possible outlier values, and to make sure they can generalize the incoming data.

There is no earlier attempt to predict ticket data like this at Elisa meaning there are no obvious guesses how the model should behave.

Next we will train different machine learning models with 75 % of the data, meaning 198 tickets. The rest, 66 tickets, are used to test how well they will get predicted. Model performances are compared in the Figure 5.11. ROC curve shows the true positive rate against the false positive rate for each model and greater area under the curve indicates better model performance.

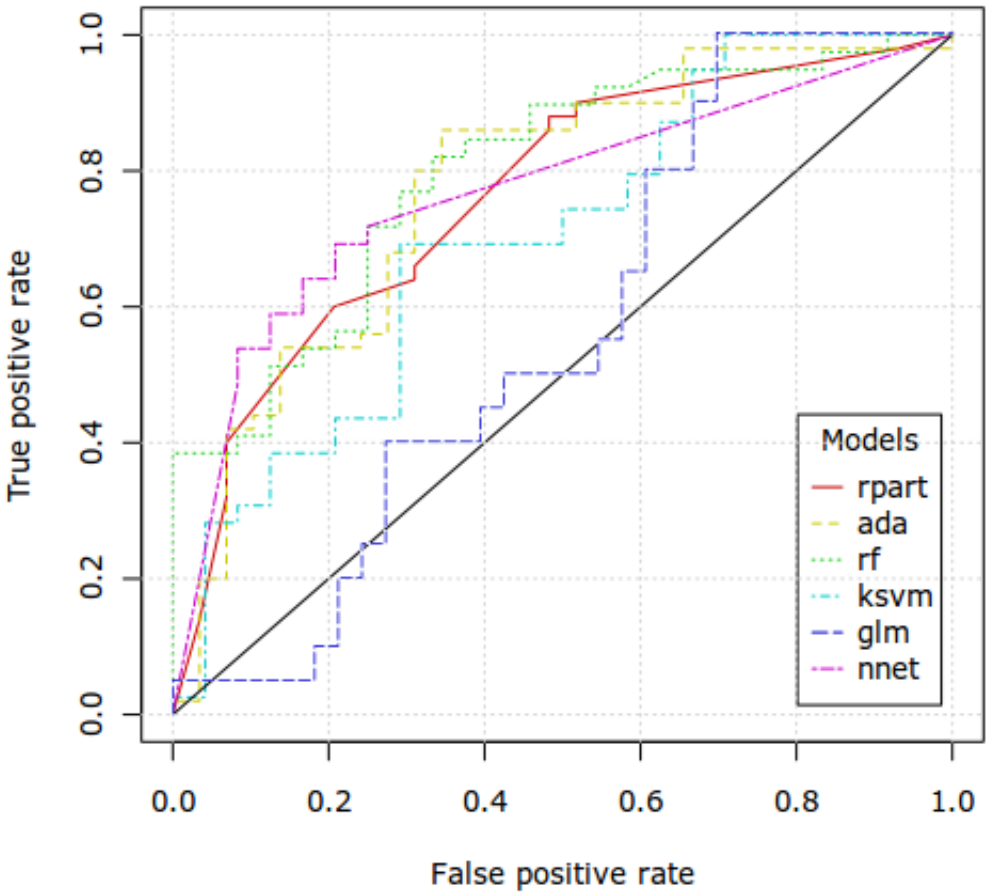


Figure 5.11: ROC plot of different learning algorithms. RandomForest (rf) shows good result for our learning task with the greatest area under the curve.

Within all the algorithms, the RandomForest (rf) model seems to do the best job with nearly 84 % area under the ROC curve. Other models are not that far from RandomForest though, but Logistic Regression (glm) has the lowest score here. The used R codes for model training are listed in Appendix A.

Next we compare if there is a difference between classification group predictions. Model's total accuracy is calculated using Equation 5.1, which

simply describes the percentage of correct predictions by dividing the True Positive (TP) and True Negative (TN) predictions with also the incorrect predictions False Positive (FP) and False Negative (FN).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.1}$$

In addition to accuracy, we will name model's specificity and sensitivity. Specificity, described in Equation 5.2, indicates the proportion of error predictions (1) that got predicted correctly. For this case, sensitivity, shown in Equation 5.3, means the proportion of no-error predictions (0) that got predicted correctly.

$$Specificity = \frac{TN}{TN + FP} \tag{5.2}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{5.3}$$

The initial model accuracy was roughly 75 %. The Random Forest model, built to compare the different algorithms generally, was created by using 500 trees and 3 randomly sampled parameters ($mtry$). The default mtry value [9] for a classification rf-model is defined in Equation 5.4.

$$mtry = \sqrt{parameters} \tag{5.4}$$

In our case, the default mtry value rounds to 3 with 13 parameter candidates. By increasing mtry to 4, but no more, we were able to tune the model to find a couple more actual error cases, bringing final model accuracy to 77.4 %.

At the same time, the increase in the number of decision trees within the forest, did not give any performance boost - actually, for this size of training data set, already 300 trees produced somewhat similar model performance, as is shown in the Figure 5.12. For our primary goal though, the forecasting of events when there is no actual error, the prediction error rate seem to stabilize to slower rate just before 500 tree nodes are reached.

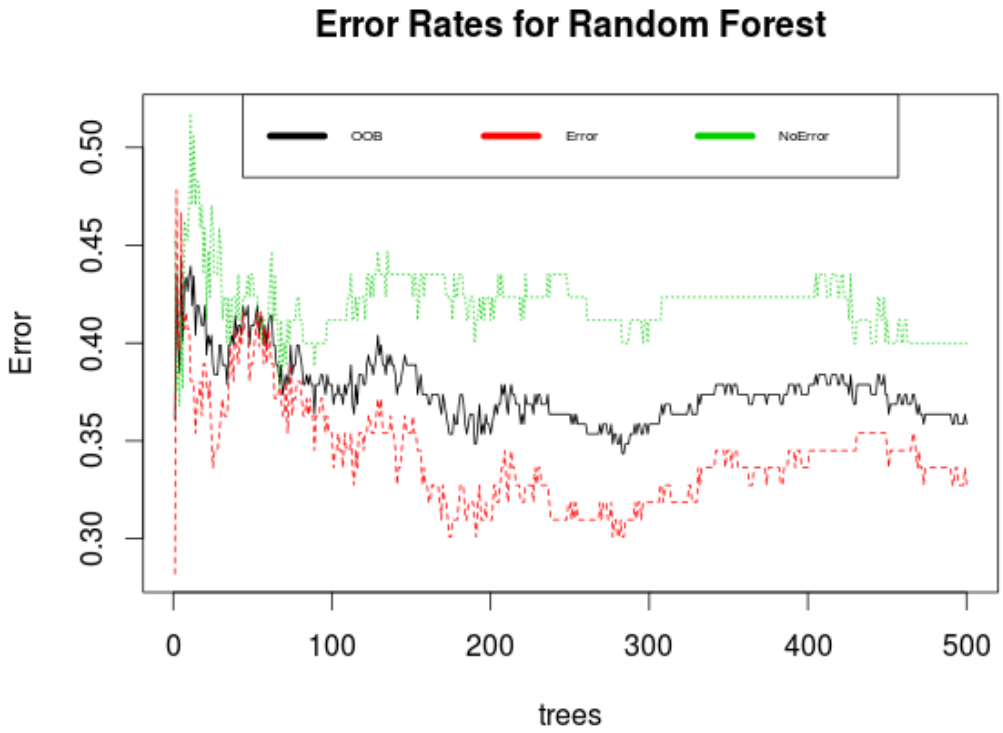


Figure 5.12: The effect on prediction model error rates when increasing the number of trees within the Random Forest.

```
Error matrix percentages for validation dataset

      Predicted
Actual  0  1
     0 28  9
     1 13 49

Model accuracy: 77.4 %

OOB estimate of error rate: 35.86 %
```

The final model had sensitivity of 68 % and specificity of 83.9 % meaning from no-error outcomes 68 % were made correctly and error outcomes were 83.9 % correct. The Out-Of-Bag error estimate for training set was 35.86 %.

Chapter 6

Evaluation

No imputation [23, p. 322] was used when there were missing feature values in EDA observations. Instead, all those observation rows were removed from analysis as we had large amount of observations. For basic statistical scenarious on big data environment, a lose of small portion of node data might not be that critical.

We used median instead of arithmetic mean to present central tendency, because there were several high error rate devices reporting every usage action as error event thus making the mean not present well the central situation. The devices with more than 200 error seconds a day were considered as outliers and they contributed approximately 5 % of all STB device observations. The error seconds contributed by those top error devices was as high as 92 % of all STB errors.

6.1 Spatial Evaluation

Instead of Finnish municipality borders, we tried to draw the actual network distribution areas to the map by using convex hull to known coordinates with a known area code. Some of the network devices were mapped to wrong area code causing the map layers reside one above another.

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.	SD
1.00	7.00	14.50	24.18	31.50	186.00	25.20

Table 6.1: Summary of kriging prediction dataset.

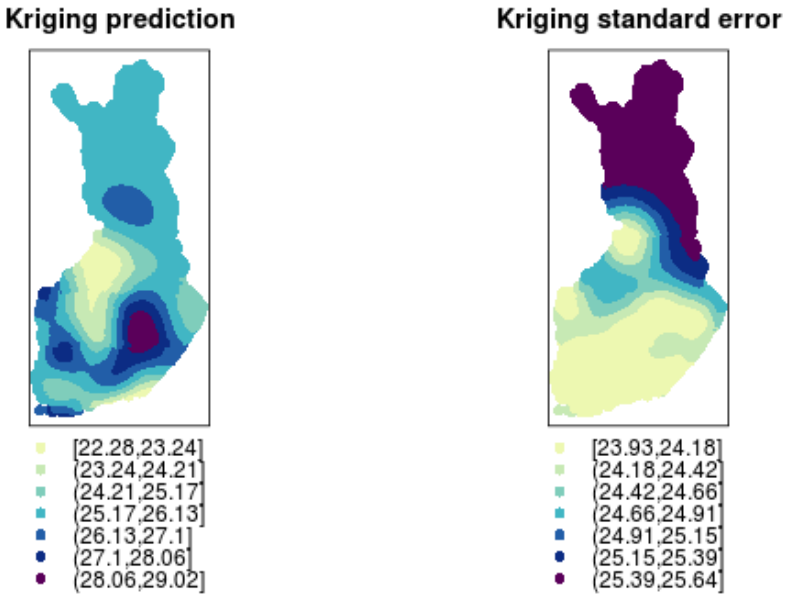


Figure 6.1: Kriging prediction and its standard error for weeks 37-39.

Kriging result consists of both predicted values and their standard error uncertainty estimates as shown in Figure 6.1. Areas with greater standard error indicate more interpolated values instead of real measurements. For example, one can see that Northern Finland has greater standard error area thus indicating smaller number of recorded actual device installations with errors. Used dataset is in Table 6.1. Device errors have a mean of 24.18 seconds a day with a stardard deviation (SD) of 25.20 seconds. When moving to the north, the error rates get closer to 0 due to SD.

6.2 Predictive Evaluation

The final tuned random forest model had accuracy of approximately 77 %. We can consider this as a fair result. Out of 66 items in the validation set, 13 were assigned incorrectly. The model used default *randomForest* package that originates from Fortran code by Leo Breiman and Adele Cutler.

Variable Importance

	MeanDecreaseAccuracy
R_STATUS	23.11
RATE_DOWN	6.08
RATE_UP	5.23
TRANSMITPOWER_DOWN	4.95
MAXRATE_UP	4.67
MAXRATE_DOWN	4.60
PROFILE	2.89
DSL_DOWN_COUNT	2.83
NOISEMARGIN_UP	2.72
ATTENUATION_UP	2.31
ATTENUATION_DOWN	2.15
TRANSMITPOWER_UP	0.72
NOISEMARGIN_DOWN	-1.83

The variable importance for the final ticket prediction model shows that the factor R_STATUS, set-up by the customer service, brings most of the model accuracy. Building a prediction model without status information brought accuracy close to 50 %, which means the model could not make clear decision whether there was a actual problem or not. If we would want to have accurate automated prediction without the need for customer service intervention, we would need more predictor variables, such as data from the ADSL modems.

Another way to improve the model would be to verify that the training data is actually representing correctly the misbehaving DSL parameters.

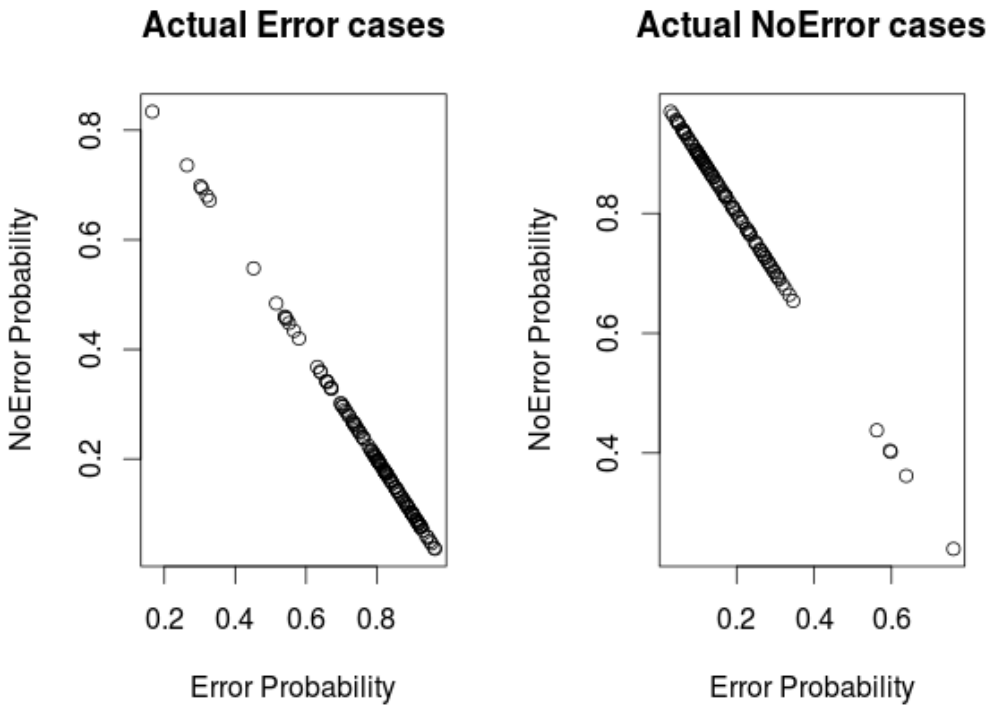


Figure 6.2: Ticket prediction probabilities from whole dataset. NoError cases seem to group up better in diagonal than the error cases.

In Figure 6.2 we can see that the *NoError* data has just a few outliers that ended up predicted as errors. At the same time, in the real error group, more items we assigned as *NoError* events. The model seems to be more confident when predicting the *NoError* cases and some accuracy is lost when error cases are predicted as NoError outcomes. By manually validating these uncertain cases, the model accuracy could be tuned further. Especially the outcomes with the highest incorrect prediction probability are interesting for the model performance.

Chapter 7

Discussion

Elisa has several individual business units focusing on their own work and own analytics. There are a number of high quality data sources for analytics, but the systems are separate and designed without thinking of a common analytics possibility, thus making it harder to retrieve data in short time. Some of the databases are located at subsidiaries requiring permission management requests that can take a lot of time to process.

There are also several different devices and platforms making the same service. For Elisa Viihde only, there are different newer and older models of STB, each having their own data format, data source and interface for reporting their behaviour. There is also a clear need to improve the error reporting agent on the analyzed STB devices.

Elisa's future focus for different business units is clear though - there is a clear interest towards better and closer to real-time analytics possibilities, but it requires work and more transparency between units. For example, STB data should be easily accessible and comparable to other parts of infrastucture, such as network data. The individual business units have the best understanding of their own data. Centralizing all the possible data under a common company-wide analytics platform might give too much power and responsibility for the individuals operating only the analytics, and not understanding enough about the required analysis target. Such a team showing a proof-of-concept analytics of other business unit's data, might not convince employees too easily. On the other hand, the domain knowledge professionals are usually working hard with the current systems thus lacking the time for thinking about new analytics solutions. Perhaps, the big data analytics should be introduced to the company by balancing between these strategies. Some analytics professionals are needed to show what is possible and how the analytics platform can be accessed, and still, some domain experts need to have time to configure the relevant pieces of data and systems, ultimately

ending up in an analytics solution for relevant case.

Even inside Elisa there were privacy problems. Direct access to a single data source was not accepted and instead, unit first cleaned their data and sent it later manually. This caused problems such as there was first no usage information, but only error data. Later some usage information was retrieved but still it did not give a clear picture for analytics and the Thesis time frame forced us to stay with the older non-complete data source.

Chapter 8

Conclusions

The Thesis concentrated on data analysis methods for problem management scenarios at telecommunication company Elisa. The analysis architecture was implemented using open source solutions such as the R programming language and Apache Hadoop. The analytical work was split to exploratory data analysis part and predictive analysis part. In the exploratory analysis, the given data was transformed into knowledge by using readable text-based statistical summaries, such as medians and means, and by using visualization methods, such boxplots, histograms, net graphs and heatmaps.

Spatial analysis was included to the exploratory part to understand better the possible differencies in service quality due to telecommunication network infrastructure. Kriging models were used to visualize Set-Top-Box error rates for different geographical areas for given times.

These findings gave the company an idea about:

- how the Set-Top-Boxes behave and which devices are affecting most the performance indicators,
- which of the worst Set-Top-Boxes might form a bad factory batch,
- how the ADSL modem firmware update affected the service quality and especially line renegotiations, and
- how the electrical copper cable measurements change on average during a year and which parameters seem the most important for predicting a working subscription.

Due to these findings, several customers were contacted and worst devices were changed and taken for further investigations. The general customer feedback for these actions was good, when the caretaking actions were performed without customer being the first one calling about a problem.

As the last part of Thesis, a machine learning model was created and evaluated to predict events when a field repair service is actually not needed thus making it possible for the comapny to save money. The performance of Random Forest, Classification Tree, AdaBoost, Support Vector Machine, Logistic Regression and Neural Network algorithms were compared, and Random Forest was chosen as the best performing model for the given dataset. The model parameters were tuned further and final prediction accuracy of approximately 77 % was achieved.

Bibliography

[1] The R Project for Statistical Computing. `http://www.r-project.org/`. Accessed Mar 2014.

[2] Apache Software Foundation. Apache Drill. `http://incubator.apache.org/drill/`. Accessed Mar 2014.

[3] Apache Software Foundation. Apache Hive TM. `http://hive.apache.org/`. Accessed Mar 2014.

[4] Apache Software Foundation. Apache Sqoop. `http://sqoop.apache.org/`. Accessed Mar 2014.

[5] Apache Software Foundation. Welcome to Apache Hadoop! `http://hadoop.apache.org/`. Accessed Mar 2014.

[6] M. Barlow. *Real-Time Big Data Analytics: Emerging Architecture*. O'Reilly Media, 2013.

[7] Luiz André Barroso, Jimmy Clidaras, and Urs Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Second Edition*. Synthesis Lectures on Computer Architecture. Morgan & Claypool Publishers, 2013.

[8] Dhruba Borthakur, Jonathan Gray, Joydeep Sen Sarma, Kannan Muthukkaruppan, Nicolas Spiegelberg, Hairong Kuang, Karthik Ranganathan, Dmytro Molkov, Aravind Menon, Samuel Rash, Rodrigo Schmidt, and Amitanand Aiyer. Apache Hadoop Goes Realtime at Facebook. In *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*, SIGMOD '11, pages 1071–1080, New York, NY, USA, 2011. ACM.

[9] Leo Breiman. Random forests. *Mach. Learn.*, 45(1):5–32, October 2001.

[10] Erik Brynjolfsson, Lorin M. Hitt, and Heekyung H. Kim. Strength in numbers: How does data-driven decisionmaking affect firm performance? *SSRN eLibrary*, 2011.

[11] David L. Cannon. *ITIL Service Operation*. TSO The Stationery Office, 2007.

[12] Mark Culp, Kjell Johnson, and George Michailides. ada: An R package for stochastic boosting. *Journal of Statistical Software*, 17(2):1–27, 9 2006.

[13] David Hand, Heikki Mannila, Padhraic Smyth. *Principles of Data Mining*. The MIT Press, 2001.

[14] R Documentation. R: Memory Limits in R. `http://stat.ethz.ch/R-manual/R-patched/library/base/html/Memory-limits.html`. Accessed Jan 2014.

[15] Gareth James, Daniela Witten. *An Introduction to Statistical Learning*. Springer, 2013.

[16] Robert Haining. *Spatial Data Analysis Theory and Practice*. Cambridge University Press, 2003.

[17] Ismo Hannula. Statistical inference for generalized additive models with an application to mothers' depression symptoms. Master's thesis, School of Information Sciences, Statistics, University of Tampere, Finland, 2011. `http://tampub.uta.fi/bitstream/handle/10024/82686/gradu05189.pdf`.

[18] Simon Haykin. *Neural Networks A Comprehensive Foundation, 2nd edition*. Prentice-Hall, Inc., 1999.

[19] Yehuda Koren. The bellkor solution to the netflix grand prize, 2009.

[20] Sergey Melnik, Andrey Gubarev, Jing Jing Long, Geoffrey Romer, Shiva Shivakumar, Matt Tolton, and Theo Vassilakis. Dremel: Interactive Analysis of Web-scale Datasets. *Proc. VLDB Endow.*, 3(1-2):330–339, September 2010.

[21] Stuart Russel, Peter Norvig. *Artificial Intelligence A Modern Approach, 3rd edition*. Pearson, 2010.

[22] A. Thusoo, J.S. Sarma, N. Jain, Zheng Shao, P. Chakka, Ning Zhang, S. Antony, Hao Liu, and R. Murthy. Hive - A petabyte scale data warehouse using Hadoop. In *Data Engineering (ICDE), 2010 IEEE 26th International Conference on*, pages 996–1005, March 2010.

[23] Trevor Hastie, Robert Tibshirani, Jerome Friedman. *The Elements of Statistical Learning, 2nd edition.* Springer, 2011.

[24] Usama Fayyad, Georges G. Grinstein, Andreas Wierse. *Information Visualization in Data Mining and Knowledge Discovery.* Academic Press, 2002.

[25] Tom White. *Hadoop - The Definitive Guide: Storage and Analysis at Internet Scale (3. ed., revised and updated).* O'Reilly, 2012.

Appendix A

R Code for Model Training

A.1 Logistic Regression

```
> ml_logit <- glm( ERRORBINARY ~ ., data=mldata,
                   family=binomial( link="logit" ) )
```

A.2 Decision Tree

```
> ml_dtree <- rpart( ERRORBINARY ~ ., data=mldata,
                     method="class",
                     parms=list( split="information" ),
                     control=rpart.control( usesurrogate=0,
                                            maxsurrogate=0 ) )
```

A.3 Adaptive Boosting

```
> ml_adab <- ada( ERRORBINARY ~ ., data=mldata,
                  control=rpart.control( maxdepth=30,
                                         cp=0.010000,
                                         minsplit=20,
                                         xval=10 ),
                  iter=50 )
```

A.4 Random Forest

```
> ml_rforest <- randomForest( as.factor( ERRORBINARY ) ~ .,
                              data=mldata, ntree=500, mtry=3,
```

```
                                importance=TRUE,
                                na.action=na.roughfix,
                                replace=FALSE )
```

A.5 Artificial Neural Network

```
> ml_ann <- nnet( as.factor( ERRORBINARY ) ~ ., data=mldata,
                  size=10, skip=TRUE, MaxNWts=10000,
                  trace=FALSE, maxit=100 )
```

A.6 Support Vector Machine

```
> ml_svm <- ksvm( as.factor( ERRORBINARY ) ~ ., data=mldata,
                  kernel="rbfdot", prob.model=TRUE )
```