Design of an Auto-generated Quote and Engineering Method of Procedure System

Akash Patel

A thesis
in The Department
of
Computer Science and Software Engineering

Presented in Partial Fulfillment of the Requirements
For the Degree of Master of Computer Science at
Concordia University
Montreal, Quebec, Canada

July 2016

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Akash Patel
Entitled: Design of an Auto-generated Quote and Engineering Method of Procedure System

and submitted in partial fulfillment of the requirements for the degree of

Master of Computer Science

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair

_____ Examiner
Dr. Leila Kosseim

_____ Examiner
Dr. Dhrubajyoti Goswami

_____ Supervisor
Dr. B.Jaumard

_____ Co-Supervisor
Dr. Marie-Jean Meurs

Approved By _____
Chair of Department or Graduate Program Director

_____ 2016 _____ _____
Name
Faculty of Engineering and Computer Science

Abstract

To be more competitive in the market, many companies are trying to speed up the quotation process,produce quote with more attractive prices, and have identified a need for support in the quotation process. The goal is to reduce the quotation lead-time, and ensure a higher level of accuracy in the estimation of the operation hours. In the case of Ciena, a first investment has been made for an automated quotation system with respect to the equipment/products that are sold by the company (currently under development). However, Ciena is also interested in including the price quotation for customer projects as well, and this is a far more challenging enterprise as it not only includes equipment/products, but services (e.g., circuit, node or ring updating or replacement) and highly specialized manpower. In addition, the execution of the services depends on the quality of the customer network and the experience and expertise of its technical support and type of equipment. The objective of the research project is to design and develop a web-based quote management application in order to semi-automate the quoting process for CIENA internal and external sales or technical engineer personnel.

Acknowledgments

I am grateful to my supervisor, Dr. Brigitte Jaumard and co-supervisor Dr. Marie-Jean Meurs for their encouraging, personal guidance and their efforts to explain things, have provided a good basis for the present thesis. It's my fortune to have them as my supervisors. I particularly appreciate that despite their enormous workload, they always made themselves available to me, and took the time to revise my thesis and papers for publications.

Next I would like to thank all the faculty members and staff of the Computer Science Department. I am grateful to the Faculty of Engineering and Computer Science, Concordia University, for supporting in part of this thesis work. Also I would like to express my deep appreciation to the members of the Committee: Professors Kosseim and Goswami for their valuable feedback on my thesis. Their comments, questions, and suggestions have been very useful to improve my work.

Lastly, I dedicate this thesis to my parents, who always encourage me and offer continuous moral support by wishing me well in my career. Their boundless love and dedication are always the inspiration throughout my life.

Contents

List of Figures

List of Tables

Acronyms

AQE Automated Quote System.

CRM Customer Relationship Management.

EMOP Engineering Methods Of Procedures.

ML Machine Learning.

MW Maintenance Window.

RAKE Rapid Automatic Keyword Extraction.

RFQ Request For Comment.

SV Support Vector.

TF-IDF Term Frequency - Inverse Document Frequency.

Chapter 1

Introduction

1.1 Project Description

To be more competitive in the market, many companies, e.g., Apttus [12], IBM [32] etc. are trying to speed up the quotation process, produce quotes with more attractive prices, and have identified a need for support in the quotation process. The goal is to reduce the quotation lead-time, and ensure a higher level of accuracy in the estimation of the operation hours. In the case of Ciena, a first investment has been made for an automated quotation system with respect to the equipment/products that are sold by the company (currently under development). However, Ciena is also interested in including the price quotation for customer projects as well, and this is a far more challenging enterprise as it not only includes equipment/products, but services (e.g., circuit, node or ring updating or replacement) and highly specialized manpower. In addition, the execution of the services depends on the quality of the

customer network and the experience and expertise of its technical support and type of equipment. The objective of the research project is to design and develop a web-based quote management application in order to semi-automate the quoting process for Ciena internal and external sales or technical engineer personnel.

1.2 Contribution of the Thesis

We are proposing a semi-automated framework incorporating several algorithms to perform the following tasks - auto quote generation, keyword extraction, feature reduction and recommend similar past quotes based on tagging for comparison purpose.

The first contribution of the proposed system is to auto-generate a new quote by applying various machine learning techniques. After having collected a large set of historical quote data for a particular type of operation, the system will trigger the decision library, which will apply a set of regression algorithms such as "Support Vector (SV) Regression", "2nd Order Polynomial Regression", "Logistic Regression" and "Linear Regression" to decide which regression algorithm provides a better fit to the dataset. The system will then build a quote accordingly. It is important to note that each quote output parameter can have a different regression model for its prediction.

Another contribution is to improve prediction of the output parameters of a quote for a specific type of operations by including or excluding input parameter into the formula responsible for predicting it. The inclusion of new input parameters is done

by a feedback module. When the system observes huge a large between the predicted value and the real value of the output parameters of a quote, it will redirect the user to the feedback module where the system accumulates a set of reasons behind discrepancy, and extract keywords from it to provide a suggestion for including a new input parameter to the user. Excluding input parameters that play a minimal role in predicting the output parameter of a quote is also an important task that improves the generalization of a prediction model. Therefore, the system applies a random forest algorithm to rank the input parameters, and let the user decides which input parameter should be removed.

Lastly, we applied an approach called tagging to quotes, which provides added information or semantic information or description about a particular quote. In our framework, we are demonstrating a usage of the Term Frequency - Inverse Document requency (TF-IDF) algorithm to calculate the similarity between queried tags and tags attached to quotes. The system returns all the quotes whose tags have high similarity with queried tags. So, a network engineer can gather all similar quotes by tags and possibly can find some help from it regarding migration details.

1.3 Plan of the Thesis

The thesis is organized as follows.

Chapter 2 presents background on existing web-based quote systems and highlights various unique features of such systems. It also details about the current quote

system at Ciena. Lastly, it summarizes problems associated with the current quote system at Ciena.

Chapter 3 includes information about various machine learning algorithms, keyword extraction algorithm and document scoring algorithm which establish the context to understanding the workflow of an overall system.

Chapter 4 is subdivided into four sections: Architecture Overview, Machine Learning (ML) Features, General Features and Technological Choices. The Architecture Overview Section provides a basic understanding of the workflow of automatic quote generation procedure including constructing the formulas for prediction and improving those formulas from the suggestion of network engineer. The ML Features Section comprises details regarding "data preprocessing", "auto-generation of quote using machine learning methods", "feedback module", "feature selection/reduction". General Features section incorporates details of "Tag Searching", "Web-based Engineering Method of Procedure (EMOP) Template Building", "Web-based EMOP building", "EMOP uploading", "Dashboard for project", "Quote version Control". The fourth Section explains the technological choices made to implement various features mentioned above. It also shows the comparison with other technologies.

In Chapter 5, we aim to simulate all the phases of the Automated Quote EMOP (AQE) system. The AQE system typically goes through three stages: initialization stage, early training stage, and steady stage. The end results of each stage are illustrated by figures. At the end, the AQE system runs a random forest ranking algorithm on specific output parameters, and shows the ranking of each input parameter

associated with it.

In Section 6.2, we begin by describing the difference between semi-automated and fully automated systems, and then discuss the set of constraints imposed by Ciena, which delayed the development and proper testing of the AQE system. In Section 6.2, we talk about the future actions to be taken in Ciena as well as possible future works related to this thesis.

Chapter 2

Background

This chapter provides background on existing quote systems, including Ciena's quotation process. It describes some drawbacks of Ciena's current quotation methodology, and the advantages an automated tool can bring.

2.1 Background on Existing Quote Systems

The quote creation process is a very time-consuming and daunting task, as it requires many input variables such as the services, prices, description, customer information, etc. In the daily quote process, the faster a quote is issued, the better are the chance of being selected by a customer. So, there are several companies in the marketplace nowadays developing quoting software to make quoting process efficient and consistent.

According to Quotewerk [8] - "Most of the companies have a Customer Relationship Management (CRM) and an accounting systems in place but, they have not yet realized the power of web based quoting system". QuoteWerk focuses on the automation of quote creation process by trying to bridge the gap between CRM and the accounting system. Spreadsheet, an Excel based manual quotation paradigm is very time consuming as the user has to fetch information about the quote from various other sources. But, QuoteWerk can be attached to various sources like an external database or Excel sheet for retrieving the information quickly, according to the given parameters, like customer name. It also has the ability to attach product images and files containing information about contracts, terms and conditions etc.

Quotient [43] is a very-intuitive online quoting software which lets the service team create, send and manage quotes. Quotient makes the quote creation process online, where the customer can accept the quote at any time and on any device. The user also has the capability to save the ideal quote template and can form a new quote out of it quickly, which leads to sales automation. Moreover, it provides visualization of the quoting data to get a better understanding of business performance.

The WorkflowMax [10] quoting software allows the customers to have an idea about how much a particular job will cost them. The list of advantages stated on their official site is as follows. First, you can create a custom quote in a matter of minutes. Second, you can customize the quote template according to the customer. Third, it provides capability for the user to attach a logo or brand information on their quote. Fourth, it gives you the control over the prices as it uses a different

markup percentage, which gives the total control over the final price.

The Axioms [3] sales manager is also a web-based sales application targeting medium-sized businesses, providing an easy and quick way to create sales quotation. It also provides the interface for easy management of the customer's sales, pricing, sales pipeline data, etc.

Apptus [13] [12] provides various features to meet client requirements such as Dynamic Configure-Price-Quote, Contract Management and Revenue Management etc. It recommends bundles, pricing and configuration based on factors like regions, organizations' top sales person and prior history. Moreover, it trains the model separately for each customer, to adapt the quotes according to their changing trends.

All the aforementioned software fetch the information from various sources based on the few input parameters which are later used to generate the quotations, thereby making the process of quotation generation easier than a manual one.

However, there are several notable differences in the above mentioned quotation software and the framework that we have developed. These software fetch the values for quotation from the database and use the templates to fill-up the quote automatically. But, there is no learning algorithm running behind the scenes to adjust the quotation dynamically based on past data. Our system recommends various output parameter values needed for the quotation, like number of hours, number of technicians, network reconfiguration time, customer support meeting time, etc. by applying various machine learning algorithms which take into the account several important factors such as historical data, customer name, type of operations, as part of the

quote generation process.

Basically, our system makes a model for each type of operation separately and trains it on the dataset and it also adapts to the newly acquired information

In contrast to Apptus [13] [12], we have a feedback loop where network engineers are able to give suggestions to improve the accuracy of the system, and minimize the discrepancy between the real and the predicted values. The model creation part of the system is flexible so that users can change the relationships of various parameters, and analyze how the system is reacting based on those changes.

2.2 Ciena's Current Quote System

The current quote system of Ciena is Request For Quote (RFQ). It is a type of document that Ciena organization asks from one or more suppliers to submit a quotation for taking care of a particular project. RFQ looks to the more detailed information about the project. For example, number of Maintenance Window (MW), Engineering Method of Procedure (EMOP) creation time, planning time etc. Sometimes, it is ideal that a supplier separates the expenses into isolated blocks as it permits the purchaser to think about various offers effectively. Mostly, the request for quotes are sent out when the services or products are standardized, and when the soliciting company has the knowledge of quantity of needed products, as it permits to compare different costs easily. For instance, if a system administration organization needs to buy 2,100 PCs with determined RAM equipment size and speed, it will request that

diverse merchants present a RFQ as the item is institutionalized, which allows an organization to find the best suit by comparing different quotes.

A quote contains different elements: equipment, services, and manpower. However, services and manpower may vary from one customer to another, depending on the health of their networking and communication equipment and services, as well as on their technical staff. Similar to most small businesses, Ciena uses Excel extensively for quotation as it is a simple tool to create, track and manage quotes.

They have a template worksheet to create a quote to maintain a unique quote format across the company. In the template, the network engineer fills up the basic information regarding the project. Then, he/she will select the customer from a drop-down or he/she will enter it manually followed by selecting the type of operations. According to the chosen type of operations, Excel macros will be activated to calculate few details in the quote to provide assistance to the network engineer who is in charge of producing the quote. These macros are hand written formulas and they are not adjusted over the time.

The advantage of using this same Excel based quote template across the organization is that the layout is consistent throughout Ciena. Quotes can be saved in Excel format or PDF format in a local machine for future reference. It is also easier to delete and modify certain details in the modified new version of the same quote and email it to the customers directly.

The sample quote has been shown in the Figures 1 and 2 below.

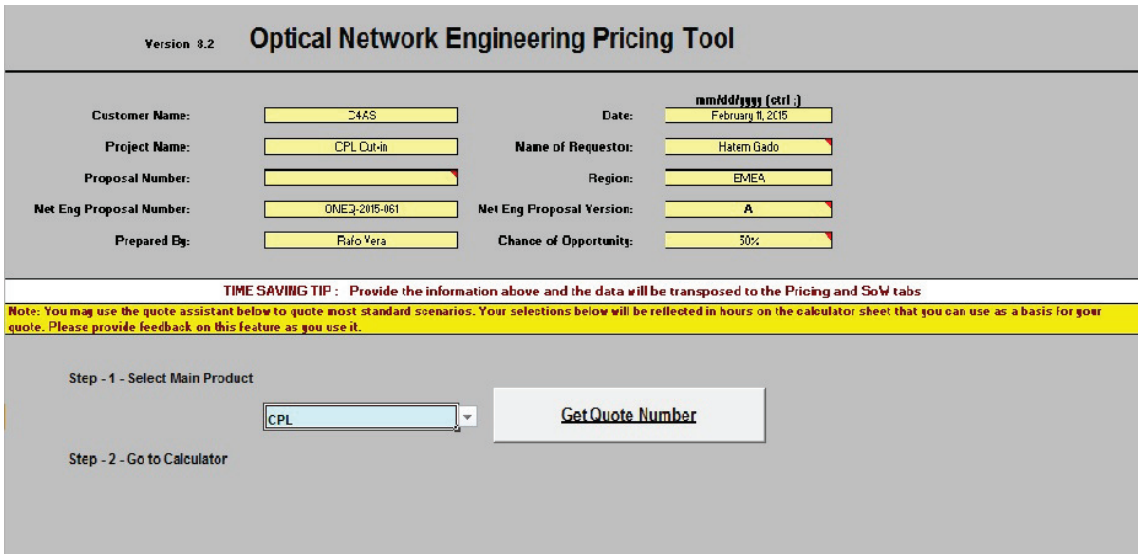

Figure 1: Project information

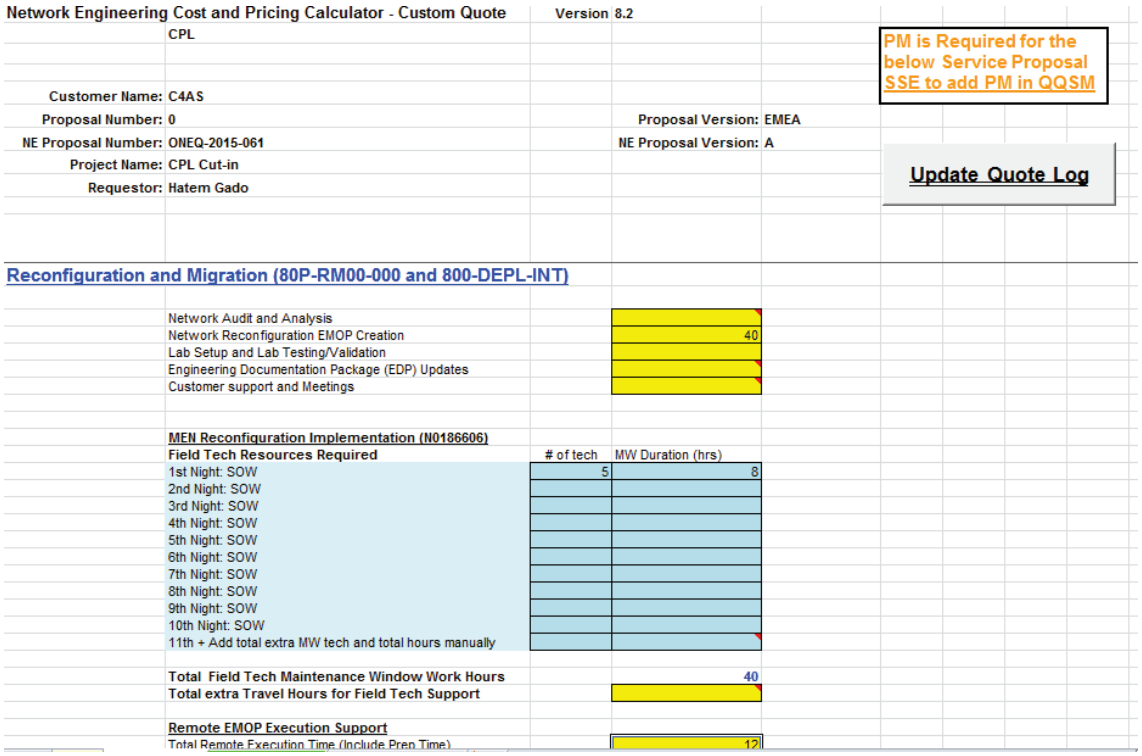

Figure 2: Detailed quote information

2.3 Need for an Automated Quote System

On the Ciena side, quotes are prepared by different groups of engineers, and some discrepancy may appear in the quotes depending on : (i) the engineer who prepares the quote, (ii) the assessment that is made on the health of the customer network, (iii) the assessment of the available information on the network connectivity and dependency, (iv) the collaboration and the expertise of the customer technical staff. Quoting styles always differ from one network engineer to another as it is highly dependent on the level of expertise of the particular network engineer who writes the quote. That, unfortunately, comes at a cost: failure in doing proper quotation will lead to failure in winning the quote or to an under/over estimation of the operation costs. More often, industries use hand written formulas to do proper quotation. But, the parameters of these formulas change over time. Therefore, static formulas should no longer be usable. So, there must be a way to adjust formulas over time.

During the quotation process, it is important to use quotes containing, e.g., a similar type of operations or network topology. Network engineers who are unfamiliar with a certain customer or type of operations face many challenges when producing a final quote. There are a large number of variables which affect the work to be quoted, and it is left to the engineer to determine the impact of each of these on the evaluation of the work to be performed. To begin working on a quote, the engineer generally gathers past similar quotations and slightly modifies them to adapt to any variation. This is not only a painstaking manual task but is also extremely susceptible

to inconsistent quotes, depending on the involved engineer.

Still, Ciena would like to have a quotation system that is as consistent as possible (reduce the price discrepancies among the Ciena engineers preparing the quotes), and as accurate as possible. On the market, there are already commercial software products, which provide web-based quote management applications in order to automate the quoting process. They are consistent in assessing the price of configured and stock items that are stored in one centralized location, with the associated manpower resources. In the context of Ciena, manpower resources can be difficult to assess as they depend on several parameters, each associated with some uncertainty.

Lastly, a few other problems attached to the Excel based quote system are described below. First,It cannot accommodate if more than one network engineer wants to access and make the changes in the same quote. Second, Excel sheet is not considered as a good database as it does not maintain relationship tables. So, it is cumbersome to integrate various details from the other related quotes. Third, the biggest hurdle to the company growth with Excel is share-ability, as Excel based quotes are tend to be made for one specific purpose e.g. node migration for MediaCom project etc. and multiple copies of the same quote are maintained throughout the organization. In the case, if it requires a slight modification on the previously created quote to make a new version of it, one has to make a new modified Excel quote and again distribute it throughout the company. So, it is hard to maintain consistency and share-ability if medium-sized or big company use Excel based quotes.

The project is twofold. One objective is to train a machine learning algorithm

on the historical data to produce quote recommendation. The second objective is to design a self-improving quote calculator so that calculated values and historical data can converge. Benefits for Ciena includes: instant feedback to engineers preparing quotes via real time guided coaching and rule validation, so that the generation time of quotes could be reduced. It is also expected that an automated system will ultimately provide a better control over prediction accuracy and optimally choose the important features and remove the unnecessary ones.

2.4 Towards an Automated Quote System

Different numbers of quotes are collected for various types of operations per year. For instance, a total of 38 quotes has been collected for the Node/Site Add/Delete operation, 36 quotes have been gathered for Node Replacement/Conversation operation and 5 quotes have been accumulated for Node Provisioning operation this year. A maximum number of quotes any type of operation usually generates, does not exceed 100 per year. So, according to our estimation, it will take atleast two years for the proposed system to gather enough quotes before making any accurate prediction.

Therefore, the focus of this thesis is on the design of semi-automated quote system. No performance evaluation will be reported as Ciena has not yet started using the system on a regular basis.

We propose an estimation of the various stages following the design of the system. We can plan three stages: Initialization Stage, Early Training Stage and Steady

Stage. In the Initialization Stage, network engineers will build relationships among input/output parameters and the proposed system will start accumulating quotes thereafter. In the Early Training Stage, after gathering enough quotes, the proposed system will select the best regression model for the data-set and build a prediction model accordingly. In the Steady Stage, the proposed system will start predicting output parameter values and accumulating feedback from each network engineer when he writes a quote to improve the prediction accuracy of the model. All steps which are performed at each stage by the system are described in Table 1.

Table 1: Stages of our system

Initialization Stage	Early Training Stage	Steady Stage
• Build an initial relationships	• Gather enough quotes (MSE or R-Square to assess it)	• Predict the output parameter value and check for discrepancy between predicted and real values
• Start accumulating set of quotes	• Select a best fit regression algorithm according to gathered dataset	• Gather the feedback from network engineers
• Revise the set of initial relationships among the input/output parameters	• Build a prediction model	• Recommend set of input parameters to network engineers to improve accuracy of their model

In addition, in Table 2, we provide an estimation of how long the system will be in each stage according to the number of quotes generated per year.

Table 2: Timeline for each stage

Quotes	Initialization Stage	Early Training Stage	Steady Stage Stabilization
• 400 quotes per year	6 months	2 months	4 months
• 600 quotes per year	5 months	2 months	5 months

Chapter 3

Literature

To provide foundations for understanding the design of a semi-automated quote generation system, this chapter first presents some basics about machine learning along with examples of machine learning algorithms. The Term Frequency-Inverse Document Frequency (TF-IDF) and Rapid Automatic Keyword Extraction (RAKE) algorithms, which are used in the system we design (see its description in Chapter 4) are then described in detail.

3.1 Machine Learning

In the domain of artificial intelligence, *machine learning* is a field that provides computational units with the capability to learn from data. Machine learning has a close connection with two other fields, namely computational statistics, and mathematical optimization. Computational statistics apply statistical methods to make predictions,

and mathematical optimization provides methods, theories and applications to machine learning. In machine learning based applications, an algorithm is trained on an input dataset. Then, the algorithm can find patterns in new data, and/or make predictions. Rather than relying on static programming instructions, machine learning algorithms improve their prediction accuracy by minimizing the (squared) error between predicted and real data [5].

One good example of machine learning algorithm is Edge Rank [29], the Facebook news feed algorithm. Facebook uses the Edge Rank algorithm to determine which article should be shown to a given user based on various factors, e.g., user interactions with friends, posts on his/her wall or likes on his/her photos, etc.

There are mainly two approaches of machine learning: supervised learning, and unsupervised learning.

3.1.1 Supervised Learning

Supervised learning algorithms [37, 25, 31, 14] construct an hypothesis function based on given labeled input datasets. Provided with an input training dataset, a supervised learning algorithm learns a model that best fits the data, and then tries to assign a label to unseen examples according to the constructed model. There are many areas where supervised learning is applied as for instance, click prediction, fraud detection, cancer detection, and sentiment classification in social media.

3.1.2 Unsupervised Learning

Unsupervised learning algorithms are a type of machine learning algorithms that try to find hidden patterns in unlabeled datasets. In a sense, unsupervised learning can be thought of as clustering the data in different groups according to pattern similarity. One example of unsupervised learning applications is market segmentation. It is the process of separating global market into a set of groups having a similarity in certain aspects, or responding similarly to market changes [28].

In this family of algorithms, there are several types of regression algorithms that are available to make a prediction. These techniques mostly involve three metrics: number of independent variables, type of dependent variables, and shape of regression line. The regression techniques we are using in our proposed system are listed below.

3.1.3 Multiple Linear Regression

In statistics, multiple linear regression constructs a regression model that intakes more than one predictor variable, and estimates a single explanatory variable. All values of X - predictor variables/independent variables - are related to the dependent variable Y.

Multiple Linear Regression is the generalization of linear regression. In linear regression [45], data are modeled using linear predictor functions, and unknown model parameters are estimated using the data. Most commonly, linear regression refers to a model in which the conditional mean of Y given the value of X is an affine function of X. Less commonly, linear regression could refer to a model in which the median,

or some other quantile of the conditional distribution of Y given X is expressed as a linear function of X. Like all forms of regression analysis, linear regression focuses on the conditional probability distribution of Y given X, rather than on the joint probability distribution of Y and X, which is the domain of multivariate analysis.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications [53]. This is because models that depend linearly on their unknown parameters are easier to fit than models which are non-linearly related to their parameters, and because the statistical properties of the resulting estimators are easier to determine.

The main goal of regression is to find the best-fitted line through the data. The best-fitted line is named *regression line*. It is computed based on the observed data by minimizing the sum of the square error of the vertical variation from each data point to line in least-square method. Since the deviations are first squared, there are no crossing out of positive and negative values.

Multiple Linear Regression is:

$$Y_i = \beta_0 + \beta_1 X_{i1} + \beta_2 X_{i2} + \ldots + \beta_p X_{ip} + \varepsilon_i \tag{1}$$

where $\{Y_i, X_{i1}, X_{i2}, ..., X_{ip}\}_{i=1}^{n}$ is the dataset of n statistical units. Y_i is the dependent variable, X_i is the p-vector of regressors and ε_i is an unobserved random variable that adds noise to the linear relationship between the dependent variable and regressors.

Two examples of multiple linear regressions are as follows:

1. Having data from brain scanners Y, and experimental design variables and confounds X, general linear regression can be applied in the analysis of multiple brain scans in scientific experiments. It is usually tested in a univariate way (usually referred to a mass-univariate in this setting) and is often referred to as statistical parametric mapping [24]

2. Multiple regression procedures are extensively applied in social and natural science research. Generally, multiple regression is used when researchers require the answer of questions like "What is the best predictor of ....?" [6]. For example, educational research wants to determine what are the best predictors of success in university. Psychologists may be curious to know which characteristics best map to social adjustment or sociologist may interested in discovering the set of social indicators that best describe that immigrant group will be able to adjust into given society or not.

However, if the relationship between predictor variables and the response variable is curvilinear, then, clearly polynomial terms are included. Therefore, to handle such cases, it is required to apply multivariate polynomial regression.

3.1.4 Multivariate Polynomial Regression

Polynomial regression is helpful when the relationship between two variables is curvilinear. Polynomial regression is a type of regression in which the relationship between

independent variables X and dependent variables Y is modeled as an n^{th} degree polynomial. It fits non-linearly related independent variables X, and explanatory condition mean of Y, denoted $E(Y|X)$. Polynomial regression can be applied to understand various phenomena such as the distribution of carbon isotope in lake sediment, growth rate of tissue and progression of disease epidemics.

Polynomial regression is a special type of linear regression. The general form of polynomial regression of the n^{th} degree is defined as :

$$Y = a_0 + a_1 X + a_2 X^2 + \cdots + a_n X^n + \varepsilon \tag{2}$$

where Y is a dependent variable, X is an independent variable, ϵ is an unobserved random error with mean zero conditioned on a scalar variable X and $\{a_0, a_1, \ldots, a_n\}$ are the unknown coefficients in the above regression function.

Because regression functions are linear in terms of unknown coefficients $\{a_0, a_1, \ldots, a_n\}$, all these models look linear from the point of view of estimation. Therefore, least square analysis, the computational and inferential problem of polynomial regression can be addressed using the multiple regression technique.

The main problem with polynomial regression is multi-colinearity. When there are more than one variable in the regression equation, then it is likely that these variables depend on each other. Therefore, it does not give a proper estimation of the curve. The solution to the problem is to map the polynomial equation to a higher

order space of independent variables called a feature space.

Some applications of polynomial regression are as follows: the execution time of programs [26], performance of a model of a sodium-cooled fast reactor [11], stock market price prediction [39].

3.1.5 Logistic Regression

Logistic regression can be considered as a special case of linear regression, but the difference between them is the value of the dependent variable, which is confined to the interval [0,1].

The hypothesis function follows a logarithm curve rather than a linear line. Logistic regression is mostly used in classification problems where values are discrete rather than continuous.

In order to understand logistic regression, one can look at the logistic function. The input value ranges from positive infinity to negative infinity while the output is a value between 0 and 1.

The logistic function is defined as follows:

$$logit[(X)] = \ln[\frac{(X)}{1-(X)}] \tag{3}$$

where

$$(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n}}{1+e^{\beta_0+\beta_1 X_1+\beta_2 X_2+...+\beta_n X_n}} \tag{4}$$

where X represents the covariates $X_1,\ X_2,\ \ldots,\ X_n$ and coefficients $\beta_1, \beta_2, \ldots, \beta_n$ represent slopes in above regression function.

Logistic regression is applied extensively in various fields, including medical and social sciences. For example, the Trauma and Injury Severity Score (TRISS), developed by Boyd *et al.* [18] to compute the patient mortality in injured patients, makes use of logistic regression. To measure injury severity of a patient, logistic regression models have been constructed by considering other medical scales [30, 17, 36, 33]. Logistic regression is also utilized to predict heart disease or diabetes considering various observed factors of the patient like blood test, age, etc. [23, 52]. Based on various features like age, income, sex, race, state of residence, votes in previous elections, it can be predicted to which party American voters will vote. Logistic regression can also be utilized in engineering to predict whether process, system or product will fail or not [51, 40].

3.1.6 Support Vector Regression

Support Vector (SV) machine can be applied to both classification and regression problems. It contains all the fundamental elements that define maximum margin algorithm : mapping inputs into high-dimensional feature spaces to perform non linear classification or regression which is so called kernel trick.

The main motivation in regression is to optimize the generalization bound. They depended on characterizing the loss function which ignores the error which are within the distance of ε from the true data points. This kind of function is so called ε-intensive loss function. Figure 3 demonstrates a case of one-dimensional linear function with ε-intensive band. Here, the variables calculate the error cost for given training points [9].

Support Vector classification or regression uses the idea of offering the solution in the manner of small subset of training points provides huge computational advantages. And by applying ε intensive function, it is satisfying two goals: assuring global minimum and optimization of reliable generalization bound [9].

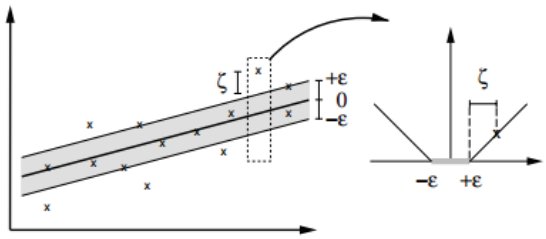

Figure 3: The soft margin loss setting corresponds for a linear SV machine

In SV regression, input set x is mapped to higher dimension by applying non linear function. And then linear models are build in this feature space. The linear model $f(x, w)$ is represented by

$$f(x, w) = \sum_{j=1}^{m} (w_i g_j(x)) + b \tag{5}$$

where, $g_j(x), j = 1, 2, .., m$ denotes the a set of nonlinear transformations, and b is

the bias term.

Basically, $L_\epsilon(y, f(x,w))$ measures the estimation quality. SV regression utilize the function so called ϵ-insensitive loss function suggested by Vapnik [21]:

$$L_\epsilon(y, f(x,w)) = \begin{cases} 0, & \text{if } |y - f(x,w)| \leq \epsilon \\ |y - f(x,w)| - \epsilon, & \text{otherwise} \end{cases} \tag{6}$$

The empirical risk is:

$$R_{emp}(w) = \frac{1}{n} \sum_{i=1}^{n} L_\epsilon(y, f(x,w)) \tag{7}$$

SV regression carries out two tasks concurrently: *(i)* minimize the model complexity by minimizing $\|w\|^2$, *(ii)* perform linear regression in higher dimension space using ϵ-intensive loss function. To calculate the variation of training samples outside ϵ-insensitive zone, there is positive slack variables $\xi_i, \xi_i^*, i = 1, 2, .., n$ has been introduced. Thus SV regression is devised as minimization of the following functional [9]:

$$\begin{aligned} &\text{Minimize, } \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{l} (\xi_i + \xi_i^*) \\ &\text{Subject to, } \begin{cases} y_i - \langle w, x_i \rangle - b, & \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i, & \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \end{aligned} \tag{8}$$

This optimization problem can converted into the dual problem and its result is

given by

$$f(x) = \sum_{i=1}^{n_{sv}} (\alpha_i - \alpha_i^*) k(x_i, x) \tag{9}$$

$$\text{Subject to,} \begin{cases} 0 \leq \alpha_i^* \leq C \\ \\ 0 \leq \alpha_i \leq C \end{cases} \tag{10}$$

where n_{sv} is the number of Support Vectors (SVs) and the kernel function

$$k(x, x_i) = \sum_{j=1}^{m} g_j(x) g_j(x_i) \tag{11}$$

The accuracy of the estimation in SV regression highly depends on parameters C, and the ϵ and the kernel parameters. The current SV regression software in marketplace takes these parameters from the user as an input and the kernel parameters and kernel functions are chosen according to the application domain.

There is the trade off between the degree to which deviation larger than ϵ are tolerated in optimization formulation and the model complexity which is decided by the parameter C. In the case of larger value of C, the main aim is to minimize the empirical risk only, without respect to model complexity in the optimization

The width of the ϵ-insensitive zone is governed by parameter ϵ, used to fit the training data. The number of the support vectors highly rely upon the value of ϵ. The smaller the ϵ, the more support vectors are chosen. The greater the value of ϵ results in a more flat estimate. Therefore, the ϵ value and C both influence the model

complexity.

The advantage of using SV regression is that the optimization problem is transformed into dual quadratic programs, and it can be used to avoid the complexity of using linear functions in the high dimensional feature space. The function is used in the case of regression penalize the error that is greater than threshold ϵ. This kind of loss functions results in a sparse representation of the decision rule which is extremely helpful in two ways - algorithmic and representational [49, 9].

3.2 Term Frequency - Inverse Document Frequency

Term Frequency - Inverse Document Frequency (TF-IDF) [34] presents how important a word is to a document as it is not only taking into account the isolated term but also the term within the document collection. It is often used in information retrieval and text mining as a weighting factor. The importance of a term increases proportionally to the number of times that term appears in a document, but is offset by the frequency of the term in the corpus.

TF-IDF is used in search engines in which a slightly modified version of the TF-IDF weighting schema is used to score or to rank a document based on the user's query. It is also applied in text summarization and classification to support stop-word filtering [50].

3.2.1 Term Frequency

From the set of documents, we want to identify sets of relevant documents for the query "the white car". An easy approach to do so is to find all the documents containing query terms "the", "white", and "car". But, still we are left with an large number of documents. For further refinement, according to the number of occurrences of the term in the document, we give a weight to that term. The easiest methodology is to assign the weight that corresponds to the occurrences of the term t in document d and sum them all together. This weighting scheme is referred to as term frequency.

$$\mathrm{tf}(t,d) = 0.5 + \frac{(0.5 \times f(t,d))}{\max\{f(t,d) : t \in d\}} \tag{12}$$

where $\mathrm{tf}(t,d)$: frequency of term t in document d, and $f(t,d)$: number of times term t occurs in document d.

3.2.2 Inverse Document Frequency

In the above example, the word "the" occurs in most of the documents. So, it will incorrectly highlight documents that contains the word "the" more frequently, without emphasizing documents containing useful words such as "brown" and "cow". Therefore, the inverse document frequency factor is integrated to lessen the effect of more frequently occurring words, and give a higher weight to rarely occurring terms:

$$\mathrm{idf}(t,D) = \frac{\log N}{|\{d \in D : t \in d\}|} \tag{13}$$

where N is the total number of documents in the corpus, and $|\{d \in D : t \in d\}|$ is the number of documents where the term t appears (i.e., $\text{tf}(t, d) \neq 0$). If the term is not in the corpus, this will lead to a division-by-zero. It is therefore customary to adjust the denominator to $1 + |\{d \in D : t \in d\}|$ [44].

3.3 Rapid Automatic Keyword Extraction

In creating Rapid Automatic Keyword Extraction [16], the main inspiration has been to build up a keyword extraction technique that is extremely efficient and can easily be applied to diverse type of documents, especially those that do not follow specific grammar convention.

RAKE considers that mostly keywords don't contain standard punctuation or stop words such as *and*, *the*, and *of*, or other such words with minimal lexical meaning.

RAKE intakes parameters such as lists of stop words, sets of phrase delimiters, and sets of word delimiters. It uses these delimiters to partition the document into candidate keywords, which are content keywords. Content keywords are the words that contain meaning within a document.

Typically keyword extraction algorithms are divided into three main phases - candidate section, adjoining keywords, and scoring and selecting keywords.

3.3.1 Candidate Section

First, The RAKE algorithm is fed with a set of word delimiters which splits the document text in the collection of its words. Then, these collections of words are again delimited into adjacent sequences at phrase delimiters and stop word positions. The Words remaining in the sequence are allocated the same position in the text, and are together considered as candidate keywords.

3.3.2 Keyword Score

After extracting candidate keywords in the first phase, a score is computed for each candidate keyword, and computed as a sum of its member word scores. The score is evaluated based on several factors: 1) Word frequency, 2) Word degree - degree of a word is the number of times a word is used by other candidate keywords 3) Ratio of degree to frequency.

The degree of a word - Deg(w) - prioritizes the word that occurs more often, and in the longer candidate keywords. Freq(w) prefers words occurring more frequently in spite of the number of words with which they co-occur. Lastly, to score individual words, the ratio deg(w)/freq(w) is used as a metric which favors the words that exist more often in longer candidate keywords. The score for each candidate keyword is computed as the sum of its member word scores.

3.3.3 Adjoining Keywords

As RAKE breaks the candidate keywords by stop words, the remaining keywords do not include interior stop words (e.g.,for, the, are, is, and etc.). Strong interest was shown in detecting keywords which carry interior stop words such as "axis of action" [16]. To discover these, RAKE looks for pairs of keywords that must adjoin each other at least twice in the same document and in the same order. The new candidate keyword is created from a combination of the keywords and interior stop words. So, the score for new candidate keywords is the sum of member keyword scores.

When done with candidate scoring, the top scored candidates are chosen as keywords for the document [38, 16].

3.4 Engineering Method of Procedure (EMOP)

When shipped to customers, every product is shipped with a product manual or documentation. The product manual contains a list of procedures for performing standard operations. Though, there are cases where customers want to re-configure their network to address growing bandwidth requirements, for equipment space consolidation activities, and for enhanced network management activities.

These customized re-configuration procedures are not addressed in standard documentation procedures as they involve multiple solutions based on the analysis of customer traffic pattern, the type of traffic they are servicing, and the budget they have

for the re-configuration. Engineering Method Of Procedure (EMOP) are the documents or procedures given to the customer based on specific customer re-configuration needs.

3.5 Qualitative vs. Quantitative Approach

Sometimes, it is important to include both qualitative and quantitative variables into regression formula for prediction of quote output parameters. For instance, the quote output parameter "number of maintenance windows (MW)" depends on the non-numeric parameter "size of customer network" (the value of "size of customer network" can be small, big or average) on top of the other numeric parameters.

Quantitative variables are observed responses that are numerical values, for instance a person's height, blood pressure, or IQ. Numerical output can also be an integer such as the number of days in a month, the number of fingers in a human hand, etc.

Values of categorical variables can be put into groups or categories. They do not necessarily have any natural ordering. Categorical variables are also called qualitative variables. An example of this type of variable is gender: it can be *male* or *female*. But, there is no ordering in the value of this variable so it is also called a nominal variable. Whereas some other categorical variable values have a natural order, for instance the common one for medical treatments: *much improved*, *somewhat improved*, *no change*. These categorical variables are so called ordinal variables.

Qualitative variables are natural fit to regression algorithms, but categorical variables should be treated differently as they cannot be added to regression models directly, and be meaningfully interpreted. If one is encoding categorical variables, for instance, network sizes with *big*=1, *average*=2 and *small*=3, and entering into the regression model then regression would look for linear effect of the network size which is not wanted. Therefore, we have to apply some coding scheme to convert this categorical variable into something meaningful that can directly be included into regression. There are many available coding scheme to encode categorical variables, for example forward distance coding, helmet coding, or backward difference coding [20].

In our Automated Quote EMOP system(AQE)(see Chapter 4 for its description), we apply a dummy coding scheme to encode the categorical variables. In practice, the k^{th} level variable will be converted into $k-1$ variables, each with two levels. For instance, a variable with 4 levels will be transformed into 3 dichotomous variables. These variables contain the same information as one categorical variable but an advantage of these variables is that they can be included into regression formula directly. This procedure of making dichotomous variables from categorical variables is called dummy coding [20].

In dummy coding, the variable is converted into two dichotomous variables. For example, business size has three level, 1=*Small*, 2=*Medium*, and 3=*Big*. This variable will be coded into two dummy variables, one called Small and other called Medium. So, when one has to code Small sized business Small value would be 1 and Medium

would be coded as 0. If business size is Medium then Small variable would be coded as 0 and Medium variable would be coded as 1. And, in last, for the Big sized business Small variable would be coded as 0 and Medium variable also would be coded as 0.

	Business Size	Small	Medium
Small	1	1	0
Medium	2	0	1
Big	3	0	0

Table 3: Dummy coding for categorial variable with three levels

Chapter 4

The Automated Quote AQE System

This chapter describes the architecture overview of the system, called AQE, which we propose. In Section 4.2, we describe its Machine Learning (ML) based features, and all its key functionality features. The ML feature described in Section 4.2, consists of three parts: data preprocessing, feedback module, and feature reduction. The general features include tag searching, EMOP template, EMOP creation, EMOP uploading, dashboard for projects and quote versioning are described in section 4.3. In Section 4.4, we discuss the different technological choices we made when we implemented the various features of the AQE system.

4.1 Architecture Overview

This section gives a high-level overview about the architecture of the AQE system. We first describe the input and output parameters of a quote and how to link output

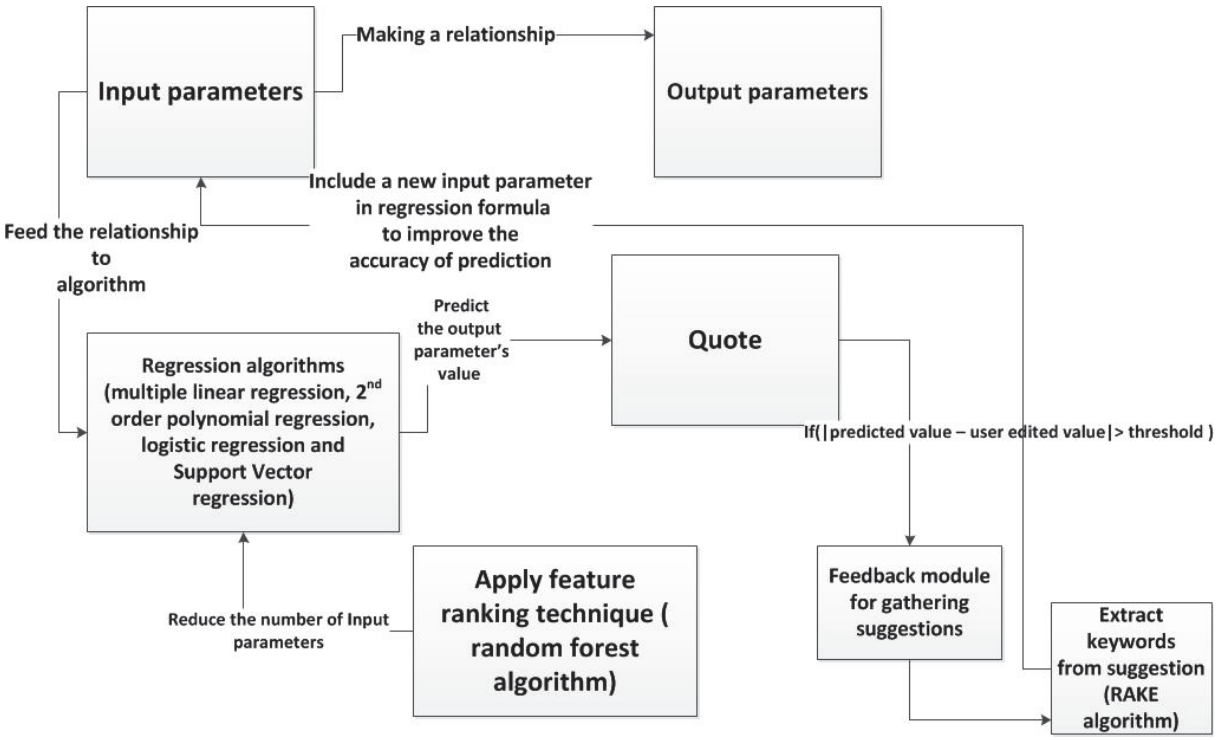

Figure 4: Architecture overview

parameters with input ones.

An illustration of the architecture is described in figure 4

Input Parameters

To create a quote with a new type of operations, a network engineer will construct a set of input questions associated with the type of operations. These input questions can be modified at a later stage, along with deletion of existing questions and addition of new questions. For example, to define a new type of operations, e.g., node migration, questions will include: *(i)* number of nodes, *(ii)* number of 2-fiber rings, *(iii)* number of 4-fiber rings, *(iv)* number of optical tributaries, and possibly some other items.

Output Parameters

The parameters that network engineers want to predict when generating the quote are defined as output parameters. Examples of output parameters are:

- **number of maintenance windows (MW)**: It is the number of days required to perform the specific type of operations.
- **network audit and analysis**: It is the number of hours required to perform audit and analysis on the customer network by a network engineer.
- **network reconfiguration and EMOP creation**: It is the number of hours needed to make a project plan by a network engineer.
- **customer support and meeting**: It is the number of hours allocated for interacting with a customer for the meeting and support purpose.

Establishing Input/Ouput Relationships

After the completion of the input question creation, the network engineer will then proceed with the relation building process. This process enables the network engineer to connect a defined set of input questions to the set of output parameters that are dependent on them. A one to many relationship can exist between the input and output parameters. There can also be dependencies among two/several output parameters.

For example, in a node migration operation, prediction of the number of maintenance windows (MW) depends on the following input questions: number of nodes,

number of optical tributaries and number of non-optical tributaries. In the relationship building process, the network engineer is then asked to associate the above input questions with any output parameters they are deemed to have an impact on.

Regression Algorithms & Quote

The necessity of defining these relationships as described above comes from the requirements to apply a set of algorithms, e.g., multiple linear regression, polynomial regression or logistic regression to predict the value of an output parameter, depending on the particular type of operation.

During the quote creation process, the network engineer will first be asked to select the operation type and to enter basic information related to the quote such as customer name, author, region etc. Then depending on the chosen operation type, the network engineer will be asked the specific set of input questions and will be guided to fill up the actual quote.

In the beginning, the AQE system starts accumulating the quote data with associated input values for building a first prediction model. Once AQE system collects an adequate number of quotes for an operation type, a suitable algorithm will be selected according to the data for quote's output parameter prediction and will build prediction model accordingly.

Feedback Module for Collecting Suggestions

For perpetual AQE system prediction accuracy enhancement, the feedback process will be triggered in two-ways. First, when the network engineer reviews the quote and does major changes in estimated quote values and second, after completing a project

when network engineers make a changes in certain parameter's value.

Whenever there is a difference between the real and the estimated values is above threshold, the network engineer will be directed to the feedback form to describe the cause for the variance. In the feedback process, a network engineer will be asked to provide suggestion describing the reason of discrepancy.

Extract Keywords from Suggestions

If a network engineer provides a reason explaining the inconsistency between the real value and the predicted values, the AQE system applies keyword extraction algorithm to extract keywords from it. Then, it lists all the extracted keywords to help the admin network engineer to formulate a new input question which will be added to the regression formula to improve prediction accuracy.

After including a new input question, the algorithm will wait for sufficiently large number of quotes being added into the AQE system to build a new regression formula. The AQE system chooses appropriate regression algorithm again to create a new regression formula to effectively predict corresponding output parameter's value.

Apply Feature Ranking Technique

To remove insignificant features or input parameters from the regression formula, we apply random forest algorithm [19] to rank the features according to their importance. It gives insight to the network engineer as how much each input parameter contributes in predicting output parameters. Then, the network engineer can remove irrelevant parameters from the regression formulas to make input parameter set more manageable.

Advanced EMOP and EMOP Template

The AQE system also supports the creation of EMOP (Engineering Method Of Procedure) templates (see Section 4.3.2 for information) and Advance EMOPs (see Section 4.3.3 for more information). Advanced EMOP and EMOP template both consists of multiple chapters or sections, including Introduction, Procedures, MW Checklist, Appendix, etc. The purpose of the EMOP template is to build a sample set of sections with variables to be defined at a later stage. An example of one of these variables could be the Customer Name, or MW number. When network engineers are looking to create new EMOPs, they can simply select the appropriate template and specify the value of the variables, and generic EMOP will be created for a network engineer. As usual, it will greatly increase efficiency and will reduce potential mistakes (incorrect procedures, spelling mistakes, etc...). The AQE system provides a web interface with a rich text editor to create the different sections of the Advanced EMOP/EMOP template. The manual effort of jumping from one EMOP to another and copy-paste certain text blocks from MS Word or Excel based EMOPs is no longer necessary. The drag and drop functionality allows the network engineer to combine various parts of template and past advanced EMOPs to quickly generate a new, custom advanced EMOP.

4.2 Machine Learning Features

4.2.1 Preprocessing the Historical Quote Data

Introduction to Data Preparation

Data preparation is one of the most critical part in machine learning as the selected algorithm learns from the dataset. Therefore, it is important to feed suitable data to machine learning algorithms. Apart from good data, one needs to make sure that one uses the proper format, and includes critical features as well. Data preparation is divided into three steps : selecting the data, preprocessing it and, sampling the data.

Selecting Data

Selecting the right data from the available pool is the first step in the data preparation. For instance, there is a wide variety of sources available online which provide diverse financial data, each with a different feature set. Hence, it is crucial to select the data with appropriate features for designing a machine learning system. The more relevant data we have, the better the AQE system will be. In our AQE system, quotes are stored in an external database. Estimation of number of hours for any type of operation is greatly different from the other type of operation. Therefore, gathering the dataset according to the network engineer selected type of operations to build the regression model is important.

Preprocessing Data

In the data processing step, the AQE system will do some preprocessing on the data so that it can be usable to machine learning algorithms. This step is further

subdivided into another three stages named formatting, cleaning and sampling. There is the possibility that different systems support different data formats. So, to make these systems work, we should feed them in its compatible format. In our case, we are fetching the data from a relational database then convert it into array type format, so that we can directly feed into the system as the regression models generated in our AQE system accept array as an input.

Another important stage in preprocessing is cleaning the data. Cleaning the data simply means removing unnecessary data or fixing missing ones. There are several cases where data instances are incomplete. Moreover, there may be missing values of some critical features in the data instances. So, it needs to be removed. For example, while making the quote, if there is problem in the server or database and the AQE system needs to be restarted in the middle of the quote making process, then the quote will be left incomplete as a result. Therefore, we cannot consider this quote as part of the learning process and we should discard such inconsistent quotes.

Sampling and Transformation of Data

For very large datasets, the running time of the algorithm will be much larger. So, the AQE system pulls the smaller sample of data instances from the larger dataset for quick exploration of the solution. This is called sampling.

Sometimes, it is required to apply some transformations on the preprocessed data. For example, our AQE system supports numeric and categorical data values. Therefore, the AQE system applies binary discretization to transform categorical data into numerical values so that we can apply regression methods directly.

4.2.2 Auto-Quote Generation Using Machine Learning

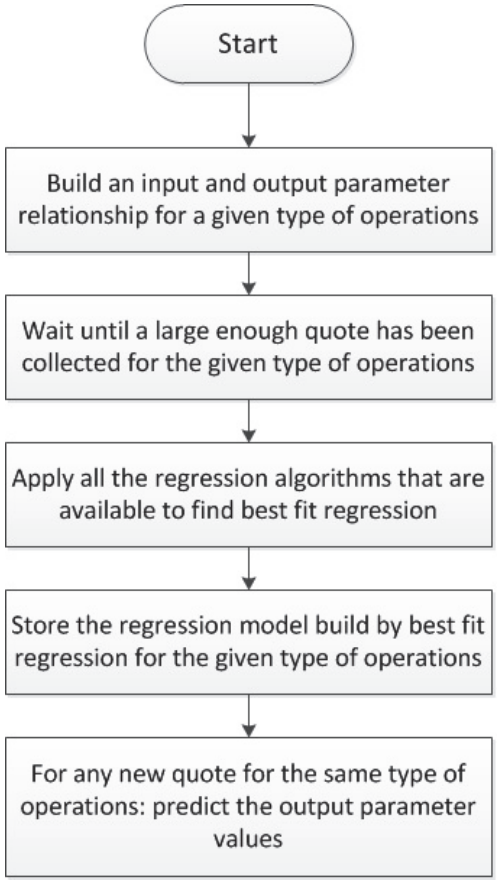

Figure 5: The workflow of auto-quote generation using machine learning

The next step after preprocessing the data is building the general learning models from that data. There are several prediction algorithms available in the field of machine learning. Regression is one supervised learning method which models the relationship between variables the relationship is refined iteratively based on the error in the prediction made by the regression model. For our AQE system, we have selected the following regression algorithms : multiple linear regression, logistic regression, 2nd order polynomial regression and support vector regression (non-linear regression) for auto-generation of quotes. The work-flow of auto-generation of quotes has been

divided into the following steps.

An illustration of the workflow of auto-quote generation is described in figure 5

Create Type of Operation

Creating a new type of operation is part of the initialization phase. First, a network engineer creates the type of operation by specifying the name, e.g., "Node Addition". Second, the AQE system will prompt network engineers to add input parameters related to the type of operation. Then the AQE system will forward the network engineer to relation-building phase where a network engineer defines the relationship between input and output parameters.

Building Input and Output Relationship

Another important part of the initialization stage is to construct relationships between the set of input and output parameters for this type of operation. In this stage, the network engineer's task is to correlate sets of input to output parameters and to feed this relationship into the AQE system for initiation. For instance, for the type of operations named "node add/delete site add/delete", the network engineer can associate output parameter named "customer support meeting time" to the set of input parameters named "number of nodes to add", "number of nodes to delete", "total number of nodes involving in the operation" etc.

Waiting for Enough Quotes to be Collected for the Type of Operations

If the number of quotes is too small, then there is possibility of noise in the dataset which can lead to wrong interpretation. It is recommended for a network engineer to wait for an adequate amount of quote-data has been collected before constructing

any regression formula. There is no clear restriction that has been imposed by the AQE system on the number of quotes it should have before building the regression model for any type of operations. However the AQE system has a network engineer modifiable variable named "Quote number threshold" for asserting a minimum quote requirement. Quote number threshold is the minimum number of quotes a AQE system must have before building any regression model for selected type of operations. For example, if the network engineer sets the quote number threshold value to 20, then the AQE system will not allow a network engineer to enter the next stage before collecting at least 20 quotes.

Apply All Available Regression Algorithms to Select Best Fit Regression

In the early training stage, the AQE system feeds quotes to the decision module, decision module is responsible for selecting the best regression for prediction for given datasets. As the AQE system is not making any assumption about the relationship between input and output parameters, i.e, whether it is linear or non-linear, it applies all regression algorithms it has: multiple linear regression, logistic regression, 2nd order polynomial regression, SV regression (non-linear regression) on the prepared dataset for the selected type of operations. Then, it will measure the mean-square error for each regression and choose which one has the lowest mean-square error. In other words, it picks the regression that better fits the dataset.

Store the Regression Model into the AQE System

After deciding on the best regression, the AQE system stores the selected regression for that type of operations in the external database. Finally, the chosen

regression algorithm will build a regression model from the provided dataset which takes a set of input data from the particular type of operations and predict the response for unknown input parameters reasonably. This generated regression model is then stored into the AQE system for later use.

Updating the Regression Model

In the steady stage of the AQE system, a network engineer might want to retrain the regression models for improving the accuracy or changing the fitting for new data. Therefore, our framework provides easily accessible interface to learning controller where the admin network engineer can visit the learning controller and run the regression algorithm again to re-optimize a prediction and make it less error free as regression always try to minimize the error rate. The admin network engineer can choose even a different regression from the last time if the AQE system finds another regression algorithm that has a better fit to the dataset.

Prediction of Output Parameters

After entering into the steady stage for certain type of the operations, e.g., node migration, when the network engineer chooses the same type of operations for making a new quote, the AQE system checks for a regression model for all output parameters related to the selected type of operations. If it finds any, it will send all related input parameters value to the regression model. Then, the regression model will calculate the final result based on the formula and will send back the final calculated result as predicted value. In summary, all the output parameters, which the regression model has already calculated, would be pre-filled in the quote.

Comparison with Apptus [13]

Apptus is building a regression model based on customers for prediction of output parameters in quotes first, then it constructs a regression model based on the type of operations. Whereas, our AQE system's flow of building a predictive regression model is completely reverse to Apptus. In addition, our AQE system is using qualitative as well as quantitative input parameters for building prediction regression model. But it is not clear whether Apptus includes qualitative input parameters into its regression models or not.

Output Parameter Independence

Each output parameter is treated separately for any type of operations. The number of required quotes for being able to start learning will be different for each output parameter. Therefore, the AQE system may be able to do some prediction earlier/later depending on the dataset for the output parameter. For instance, two output parameters number of Maintenance Windows (MW) and customer meeting time, in type of operations named site addition/deletion can be captured in a different set of quotes. Hence, each output parameter's value starts getting predicted irrespective of the others based on their quote dataset.

Forwarding to Feedback Module

All the output variables in the quote are editable so the network engineer can change the predicted value to whatever he/she finds suitable in the quote. Hence, when the network engineer saves the quote, quote-saving module will find the difference between the predicted and network engineers' assigned value if this margin is

bigger than a given threshold. This incident will be reported in the database and the network engineer will be dispatched to the feedback module to gather the possible reason of discrepancy.

4.2.3 Feedback Module

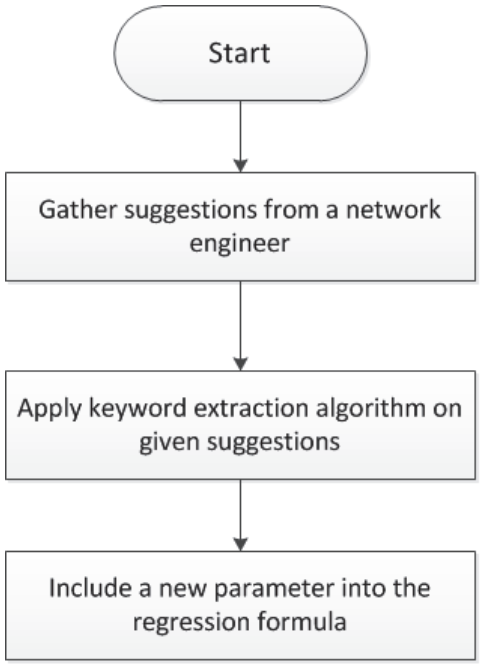

Figure 6: The workflow of feedback module

Figure 6 describes the workflow of feedback module.

Feedback process consists of several steps: gather the suggestions from a network engineer to reduce prediction error margin in output parameter's value, extracting a keyword from the suggestion for helping to build a new input parameter, adding a new input parameter into the regression formula to improve prediction accuracy.

The AQE system incrementally accumulates the batch of the quotes and regression algorithms aim for minimizing the generalization error in prediction. If AQE system

continuously suffers from significant margin of error between real value and predicted value, then it can be the case that the network engineer forgot to relate significant input parameter to the output parameters in regression. Therefore, after noticing the difference between real and predicated value beyond the threshold times, the AQE system will transfer the network engineer to the feedback module for adding a new parameter [22].

Gather Suggestions from Network Engineer

First, the AQE system accumulates all output parameters whose prediction suffers from a big margin of error. Then, AQE system will redirect the network engineer to feedback module for gathering inputs from the network engineer. Next, the feedback module will prompt the suggestion box for each output parameter which has a notable discrepancy in their prediction. Therein, the network engineer can enter a possible reason describing the discrepancy. There can be two types of suggestions. First, it can be properly formatted input parameter which we can be included directly into the regression formula. Second, it can be a long description of the possible reason for the discrepancy. In the second case, AQE system has to perform a keyword extraction operation to ease the task of constructing a new input parameter for the network engineer.

Apply Keyword Extraction Algorithm on Suggestions

Keyword extraction is a very important task here as the network engineer can quickly identify the input parameters that is worth adding or not. These keywords describe the main topic in the suggestion so it is easier for the network engineer to

determine a relevant input parameter to be added to the regression relation. AQE system will apply the RAKE keyword extraction algorithm [16] which is unsupervised, domain-independent and language-independent method of extracting the keyword out of a document or suggestion box. First, RAKE algorithm parses the text into a set of candidate keywords which are delimited by word limiters and generate candidate keywords. Second, it will calculate the score of each candidate keyword as the sum of member's score. Third, it will sort the set of candidate keyword according to its score. Thereafter, when the admin visits the input parameter modification page attached to the type of operations, they will be able to see the extracted keyword as a suggestion to add the new question to improve the accuracy of prediction of attached output parameter.

Adding a New Input Parameter

The network engineer will pick up the set of keywords which are relevant and then format the new input parameter properly so that it can be added directly into the regression equation. AQE system allows the inclusion of both qualitative and quantitative types of input parameters into regression. Whenever the network engineer inserts a new input parameter to any output parameter, AQE will stop the prediction for that output parameter and remove the related regression model. Next, it will wait for enough quotes to be accumulated with this newly included input parameter and then apply regression again to build another version of the regression model for output parameter prediction.

4.2.4 Feature Selection and Reduction

Introduction to Feature Selection

Feature selection is the process of choosing the most discriminating features out of a pool of features. The important goal of the feature selection is removing unnecessary or redundant features without the loss of relevant information. Feature selection is quite different from dimensionality reduction. The purpose of both methods is the same but they approach the problem differently. Dimension reduction method generates a new combination of the attributes, whereas feature selection method includes or excludes the parameters from the data without doing any modification.

Advantages of Feature Selection

There are two main advantages of feature selection. First, it helps the AQE system to create an accurate regression model as it selects the features that provide good accuracy whilst needing fewer data. Second, feature selection method detects and removes irrelevant attributes which contribute less to the accuracy of regression model or otherwise decrease the accuracy of the regression model.

Various Methods of Feature Selection

There are a quite number of methods available for understanding the relation between features and the response variable. Pearson correlation coefficient [15] models the linear relationship between attributes and the response variable. However, it is not adequate to use if attributes are non-linearly correlated with the response variable. Hence, for modeling non-linear relationship, tree-based methods like random forest

are used. Another advantage of random forest is that it does not need much tuning.

Problem Feature Selection Tries to Solve

In our AQE system, network engineers are allowed to add the set of input parameters associated with any output parameter to improve the accuracy of the regression model. But, it has a serious disadvantage that adding more input parameters to improve the accuracy results in more than necessary parameters and it would be unrealistic most of the time for the network engineer to fill up all input parameters for predicting output parameters of quotes

Applying Random Forest Algorithm for Feature Ranking

A network engineer chooses the output parameter on which he/she wants to apply feature selection to exclude relating few input parameters. Then the AQE system applies the random forest method to assign the score to each input parameter related to chosen output parameter. These ranked attributes are then either removed or kept from the dataset by network engineers. A network engineer can delete any number of input parameters related to the output parameter. It is required to retrain the module once a network engineer does some modification. Therefore, as soon as the network engineer deletes any of the features, the AQE system will run the decision library which in turn finds the best regression fit to the new dataset. Then, it will run the regression algorithm again with the new feature set to generate new regression model. After training the regression model with the new feature set, AQE will replace the new regression model with the previous regression model. Thereafter, the network engineer will not be asked for the deleted input parameter as the part of

quote generation process.

4.3 General Features of System

4.3.1 Tag Searching

Introduction to Tagging

Tagging is as an effective strategy which empowers a user to discover, sort out and comprehend entities. Tagging in AQE fills two needs - recommendation, and prediction. Since tags are made by the user, they speak to ideas important to them as tags are effectively comprehended by users; tags are useful in revealing the obscure relationship between items and the user. In a few occasions, it is demonstrated that specific tags that users find and settle on choices about specific things [46]. For example, there is a user named Bob who is a participant of the movie recommendation community. He is a big fan of action movies and rated four stars to the movie named "Die Hard", "The Avengers" and "First Blood". If the same person visits the webpage of "Deadpool", he would like watching it as it is tagged as action movie by another user from the community (the predicted task). Bob might visit the tag "20th Century Fox" to find a more similar movie like "The Revenant" (the recommend task) [47].

Collaborative Tagging

Collaborative tagging systems have become in high demand for sharing and organizing web resources these days, which leads to an enormous amount of user-generated tags. Social bookmarking system - del.ici.ous use tags to conceptualize, categorize,

or sharing resources. Users can attach tags to any resources on the Internet based on their sense of value [48]. With this sense of value, one can find individual and common interests among unknown users.

Tagging in Our System

In our AQE system, resources are Quotes, EMOPs, and Apps. In order to facilitate searching through these resources, a tagging system has been put in place. A network engineer is allowed to attach any tag freely and subjectively according to their sense of value. With attached tags, it is easy to find these resources, and organize them in a manner that they can be useful to the company.

For instance, in our AQE system, a quote creator or the network engineer involved in the same project are both allowed to attach any tag to the quote to provide more semantic information. They can even modify or attach more tags to the quotes later on. Moreover, our AQE system also adds another layer of default tags to the quote including project name, author, requestor, customer name, region and type of operation. The result is a final set of tags for each quote, which is a combination of system defaults and user defined.

Search relevance is a measurement of how closely related a document is to a query. Search relevance can be determined in a wide variety of ways, ranging from simple binary relevance to a weighted relevance algorithm such as TF-IDF, which assigns a relevance score to documents. It is important to note that all these techniques are used to weight terms in the query, so that more useful terms are strengthened over less useful terms. This means that documents with more mentions of higher weighted

terms will be retrieved over documents with the same number of mentions of a lower weighted term.

To find similar quotes to the current quote, AQE system compares the tags from the current quote with the tags associated with the set of quotes which are stored into the AQE system. It uses the TF-IDF technique (see Section 3.2) to score all quotes in the AQE system against the current quote to find the relevant/similar quotes. Similarly, the AQE system allows finding all similar EMOPs, advanced EMOPs (see Section 4.3.3) and applications (e.g., topology discovery app, health predictor app etc.) based on TF-IDF.

In the project summary page, the AQE system displays all the quotes and EMOP that are relevant to a particular project based on the tags attached to the project.

4.3.2 EMOP Template

Templates allow network engineers to build general content easily. EMOP templates consists of multiple sections, e.g., Introduction, Contacts, Procedures etc. Each of these sections can have multiple subsections. The development of EMOP templates for the organization will boost flexibility, provide consistency, and reduce effort, time and cost of generating a new EMOP. When Engineers work off an EMOP template. Standardization is the key to maintaining uniformity in the organization which helps other network engineers know how to create or use EMOP templates.

EMOP template creation starts with defining its name and description. Then, the network engineer specifies the name of the section to be built. Next, the AQE system

will redirect the network engineer to content creation module where the network engineer can attach multiple subsections to the section being built. Each subsection consists of three parts: the title, description, and content area. The content area of the section is equipped with a rich text editor which brings the word processor capability to the web. So, a network engineer can include text, images, tables, and can also apply styling to the content to improve the appearance. The key feature of the EMOP template is being able to define a variable in the content of the subsection. For example, in the introduction, there may be a sentence stating the customer name as well as the names of various network elements involved in the procedure. By creating dynamic variables for these two pieces of information, it allows the same procedure to be easily adapted by the Network Engineer to suit the needs of a specific project. With the adoption of EMOP templates, the organization will become inherently more standardized and time and effort of creating new procedure documentation will be reduced.

Currently, engineers at Ciena often create their own individual EMOP templates. These are stored on a user's local machine and range in complexity and detail. The documents are small pieces of work that have been improved upon over time by individuals, but not necessarily shared with the organization to promote collaboration and consistency among engineers. The AQE System aims to promote knowledge sharing by making these EMOP Templates easily accessible. They are stored in an external database which can be searched using the template name or type of operations. As EMOP Templates are saved in a centralized place, any network engineer can make

a modification and changes will be reflected instantly for all engineers to use in the future.

It also reduces the EMOP construction time as a network engineer does not need to re-create an EMOP for the same type of operation from scratch. By choosing an EMOP Template corresponding to the type of operation and defining the value of the set of variables, the AQE system will generate basic content blocks of an EMOP for network engineer in less time. With the template as a base, network engineers can customize the specific EMOP according to their detailed project requirements.

4.3.3 Advanced EMOP Creation

Introduction to EMOP

An EMOP is the documentation of an Engineering Method Of Procedure, which contains detailed information and steps an engineer should follow to modify a customer's network during a scheduled maintenance window (MW).

Old EMOP System vs. New EMOP System

We completely redesigned the traditional EMOP system as it was not the efficient one according to Ciena's employee. There are two disadvantages of the old Excel-based EMOP. First, the details of each subsection are not standardized among network engineers and makes the re-use of information cumbersome, requiring lots of manual effort to re-organize and re-format as needed. Second, EMOPs are generally kept on individual's computers and file systems. There is no easy way for engineers to search for past procedure documentation and samples. This greatly limits the

feasibility of knowledge sharing across the organization.

Workflow of Advanced EMOP Creation

Web-based EMOP creation is a very simple process. It begins with selecting a project to which one wants to attach an EMOP. A network engineer can attach multiple EMOPs to a project, one for each specific maintenance window (MW) activity. Then, the AQE system asks the network engineer to enter the name of the EMOP and its description. Next, the AQE system will fetch all similar or related EMOP templates and EMOPs from the AQE system. They will be displayed to the user in a tabular form. The network engineer is allowed to select one EMOP template and more than one past EMOP from the list. The new EMOP will be auto populated with the available content from a selected template. Then, the AQE system will find the variables that reside in that chapter/sections and will display the list to the network engineer for his input. After feeding the value in the variables, the AQE system will reflect the variable assignments in the content in subsections.

New Feature Called "Block" to Extend the Share-ability of EMOP's Content

While the EMOP template serves as a base to populate general content for the new EMOP, there are times when the user would like to include specific information that is the same as a past EMOP. If the user would like to reuse a specific section of a past EMOP for the new project, they can select this section and drag it into the desired content area. For example, in a previous project, an engineer may have made relatively small changes to a procedure and felt they were small enough and did not

merit the creation of a whole new template. If the engineer finds they want to use this slightly modified content at a later date, they can click and drag the subsection from the past EMOP into the new one.

Moreover, a new feature called "block" has been implemented into the AQE system to increase the shareability of EMOP's content. Blocks are subsections created in EMOP's chapters/sections. For example, while building a chapter/section named "procedure", the network engineer includes three subsections - "sub-procedure1, sub-procedure2, and sub-procedure3". These created subsections are considered blocks and network engineer can use these blocks in other EMOPs. So, once a network engineer selects the list of past EMOPs for creating a new EMOP, all selected EMOPs' block would be available to the network engineer as a part of new EMOP creation process. In contrast, Ciena's current EMOP creation methodology makes it quite cumbersome to re-use these procedures. The network engineer has to manually look for the section in other EMOPs. The AQE system greatly alleviates the burden of searching through old files and copy-pasting specific parts.

Tags to Advanced EMOP

Network engineers have the ability to attach tags to an advanced EMOP as it provides semantic information which describes the advanced EMOP's content. It also enables the AQE system to recommend the set of advanced EMOPs based on the tags a particular advanced EMOP is carrying. The specifically addresses the second fall back of Ciena's current EMOP creation methodology, where once complete, EMOPs are filed away on an individual's personal computer and not widely available to the

organization. Storing templates and EMOPs in a central location with searchable functionality was a critical component of the AQE System.

4.3.4 EMOP Uploading

At Ciena, network engineers develop EMOPs in Excel sheets. There are huge amounts of valuable, documented EMOPs across Ciena which are Excel based. Therefore, to maintain backward compatibility, I have implemented an EMOP uploading module in AQE where network engineers can attach EMOP with any project by providing basic information (e.g., EMOP name, Author) and attach relevant tags to it. Although the advanced-features of the web-based EMOP, such as block re-use are not available, the uploaded EMOPs still appear in any searches performed. The ability to attach tags to the uploaded files helps to foster the knowledge sharing through the use of past documents.

4.3.5 Dashboard for Projects

A dashboard is a visual display of most relevant information needed to accomplish one or more objectives. Generally the goal is to fit all this information in a single computer screen. Our dashboard aim is to combine information from various modules into a single entity and display it to the network engineer for ease of use.

The dashboard is divided into several parts as follows.

My Quotes and My Projects

On a Web site, personalization is the process of tailoring pages to an individual's

specific requirements or situation. In the context of the Ciena's network engineers, the AQE System customizes the dashboard view to suit the needs of the logged in user. All quotes and projects created by the current user or network engineer are displayed in tabular format for easy viewing. The tables will display only short summaries of the quote or projects ,e.g., quote number, project name, region etc. The row on the table also contains a link that expands when clicked, showing an extended summary of the quote or the project.

All Quotes and All Projects

While the initial dashboard view is customized to show projects relevant to the logged in user, there is still the option to view all quotes and projects in the system. By toggling this button, the network engineer can quickly switch between the extended list of information in the system and information relevant to them. While the goal of the AQE system is to make the quotation process easier and more efficient, by allowing the engineers access to all quotes and projects, it removes any limitation on knowledge sharing.

Basic Filtering

At the top of the page, there is a filter field. This filter is responsible for quickly finding the subset of rows which contain text that match what is written in the filter. For instance, if a network Engineer types the test "Node Migration" in the filter field, the AQE system will display all rows with matching information, and clears all other rows from the screen.

Tag Searching Module

Rather than using the basic filter, engineers also have the option to search using the advanced tagging system put in place through the AQE. Engineers can search related quotes, EMOPs and advanced EMOPs by tags and the results will display in descending order according to similarity and relevance.

Advanced Searching Module

If the engineer selects to use the Advanced Search option, they have more control over the specific query fired to the system. When a user wants to narrow their search to a specific kind of document or field, this module is a good option to use.Network Engineers can choose to search quotes by quote name, quote number, projects by project name and EMOP by EMOP number and EMOP name etc.

Advanced EMOP

When a project is selected, an engineer can choose to generate an Advanced EMOP for the selected project. Linking this interface to the dashboard gives quick access to a very useful module right from the front page of the AQE System.

General Upload

In order to maintain flexibility for the engineer, this online AQE system will allow a network engineer to upload and attach any document in any format to any project. It will also allow the attachment of more than one document to any project with tags. The result is an easily searchable and adaptable system which is largely governed by the tags defined by users.

Configure the Type of Operations

This section contains references to several modules. For instance, the following

modules are accessible from this section:

- "Learning Controller" where a network engineer can run the regression algorithm
- "Create a new type of operation", where relevant parameters are established
- "Modify Relation to I/O" where a network engineer can modify the relationship between input and output parameters
- "Modify Output Threshold" where a network engineer can adjust the allowed discrepancy between predicted value and real value for specific output parameter

4.3.6 Quote Versioning

The AQE system allows the management of changes in the quotes. Changes can be anything from basic quote information to increases in the number of maintenance window (MW). Every change in the quote is identified by a version number. The first copy of the quote is identified by the term "version number 1". Every change in the quote is assigned a new version number. In the quote summary page, the AQE system displays the list of all versions of a particular quote with detailed information about the most recent version. If a network engineer needs to see a past version, he/she can switch to any version listed in quote summary page and get detailed information about the selected quote version.

4.4 Technological Choices

4.4.1 Multiple Linear Regression

It is a method for modeling the linear relationship between dependent values to one or more explanatory value. To solve linear regression, there are two methods commonly used 1) Normal Equation and 2) Gradient Descent.

Gradient decent algorithm functions by taking in initial guess of the parameters and further iteratively refining the value of these parameters in order to find the best fit for a given dataset. The normal equation uses a very different approach than gradient decent as it tries to find out the parameter values in one step by computing the matrix inversion.

However, there is one major advantage of the normal equation over the gradient decent method : it calculates parameters in one step as opposed to the iterative approach of the latter method which also requires a stopping criterion to find the optimal value of the parameters. But, gradient descent works well even with a large amount of samples compared to normal equation. Indeed, normal equation requires computing the inverse of the matrix which is expensive in terms of computation cost.

I have implemented the normal equation based multiple linear regression solution in Java.

4.4.2 Other Regression Algorithms

It can be a mistake to assume the regression model for quote generation is linear. Almost all linear regression methods suffer from the drawback that many systems are non-linear. If linear regression model will be used to fit curve $y = 1 - x^2$, it will produce a very terrible result. So, to overcome such a drawback, we are applying various non-linear regression algorithms such as 2nd order polynomial regression, logistic regression and support vector regression to better fit the non-linear dataset.

We utilize python library named scikit-learn [1] to implement all non-linear regression algorithms.

4.4.3 Random Forest for Feature Reduction

For univariate feature selection which inspects every feature to strength relationship of the feature with response variable, there are several methods available like Pearson correlation, distance correlation, and maximal information coefficient. Pearson correlation coefficient is one of the easiest methods for finding the linear relation between feature variable and the response variable. Pearson correlation is similar to standardize regression coefficient that is used in linear regression. But, if the relationship between the feature and the dependent variable is non-linear, then it would give disastrous results. As an alternative, there are tree-based methods, e.g., decision tree and random forest [35]. It handles a non-linear relationship without requiring much of tuning. Therefore, we are using tree based technique named Random forest for feature ranking.

I mainly utilize a module called RandomForestRegressor from the python library scikit-learn [1] [4].

4.4.4 TF-IDF

TF-IDF [44] integrates the definitions of terms frequency and inverse document frequency to produce a composite weight for each term in each document. Here, each document can be seen as a vector with one component corresponding to each term and the value of these components is the multiplication of term frequency and inverse document frequency. To sum up, using search query, we are building up one document and trying to find similarity between other documents created from quotes and EMOPs. Then, ranking the documents according to the similarity factor in descending order.

For tag searching, we apply numeric statistic named TF-IDF from the Apache Lucene Java library [2].

4.4.5 Rapid Keyword Extraction Algorithm (RAKE)

For evaluation of the performance between RAKE against other keyword extraction algorithm described in Mihalcea and Tarau (Text Rank) [38] and supervised learning (Hulth 2003) [27], the set of 2000 Inspects abstract journal papers from Computer science and Technology have been utilized where abstracts separated into 1000 training set, 500 validation set and 500 test set. RAKE algorithm outperforms other algorithm in terms of precision and F-measure with a generated stop list based keyword

adjacency and also yields comparable recall.

We are using the python implementation of the RAKE algorithm by Aneesha [7] for keyword extraction purpose [16].

4.5 Machine Learning Approach for AQE System

In real-life machine learning problems, the first step is to extract the set of features from the input dataset which effectively and sufficiently represent the data then input data is divided into three set - training dataset, validation dataset, and testing dataset.

In a training phase, one represents their data as gold standards and train their model, by pairing input with expected output value. In order to estimate how well the model has been trained, we use validation data set to tune the parameter for reducing the risk of overfitting and then test the performance of our model on the unseen test dataset. [41]

But, in our AQE system, we do not have enough training dataset and to gather enough training dataset, it will take a while as engineers have not started using the system on a regular basis. As the network engineers at Ciena wanted the prediction to work as soon as possible without focusing more on the accuracy, therefore AQE system is designed in such a way that all the phases mentioned above can pass through with small dataset affecting accuracy of prediction.

Chapter 5

Implementation of AQE System

In this chapter, we attempt to simulate all the phases of the system on a small dataset. In order to do so, we have described all three stages in the detail. In section 5.2, we talk about the steps to follow, to initialize the AQE system. Then, we describe the early training stage in detail where the AQE system gathers enough number of quotes and chooses the best regression algorithm for building the regression model in section 5.3. At the last, we discuss the steady stage where the proposed AQE system predicts the output parameters and simultaneously accumulates suggestions from the network engineer to improve the accuracy of the regression model.

5.1 Proof of Concept

We have implemented the overall system from the scratch. All the modules, namely learning module, feedback module, and keyword extraction modules are properly

integrated into the system. All the algorithms selected for each module are according to the requirements of Ciena. A different number of quotes are collected from various types of operation. We have set up the simulation of all the module on 38 quotes collected for the type of operation named "Node/site Add/Delete". In the section below, we have described each phase of the AQE system in detail.

5.2 Initialization Stage

Network engineers at Ciena want to predict the value of four output parameters of the "node/site add/delete" operation : 1) number of maintenance windows (MW) 2) network audit analysis 3) network reconfiguration EMOP creation 4) customer support and meeting.

This operation also consists of four input parameters: 1) number of nodes to add 2) number of nodes to delete 3) number of sites to add or delete 4) number of systems involved.

5.2.1 Building an Initial Relationship

The relation between the input and output parameters are as follows: The "number of maintenance windows (MW)" output parameter is dependent on all four input parameters such as : "number of nodes to add", "number of nodes to delete", "number of sites to add or delete" and "number of systems involved". The "network audit analysis" output parameter is related to the same input parameters as mentioned

above and also related to the output parameter named "number of maintenance windows (MW)". But, "network reconfiguration EMOP creation" output parameter is dependent on "number of maintenance windows (MW)" output parameter and other three input parameters such as: "number of nodes to add", "number of nodes to delete" and "number of systems involved". At the last, "customer support and meeting" output parameter is related to the above three output parameters named "number of maintenance windows (MW)", "network audit analysis" and "network reconfiguration EMOP creation".

5.2.2 Start Accumulating the Set of Quotes

After building the relationship between the input and output parameters for the type of operations, the proposed AQE system will start collecting quotes related to it.

5.2.3 Revise the Set of Initial Relationships

Network engineers are allowed to change the initial set of relationships among input/output parameters for getting a desired effect in the prediction of particular output parameter after the proposed system passes through all the phases. Therefore, whenever the network engineer changes the set of relationships, the system should build a new regression model according to the newly defined relationship. Hence, the proposed system will re-enter into early training stage.

5.3 Early Training Stage

The admin network engineer is allowed to visit the learning controller module and be able to run the decision library in order to find the best fit regression algorithm for each output parameter.

5.3.1 Gather Enough Quotes

There is a threshold value set by the network engineer for each output parameter which imposes requirements on minimum number of quotes the system must have before building a regression model. The screenshot below displays the percentage of quotes when the system has gathered 40 quotes. With this, the network engineer can have an estimation about how much more quotes are required for each output parameter.

It is to be noted that due to the inconsistencies in the quotes some output parameters may have enough values, whereas other parameters might not. For instance, in the Figure, "number of MW (maintenance windows)" shows 52 percentage complete whereas for the other output parameters, it shows 62 percentage finished.

Then, in figure 8, we show the screen shot taken after feeding more number of quotes than the threshold. It displays a message that the algorithm is ready for prediction. This time the system will apply the decision library on datasets and fetches the best fit regression for each output parameter and then stores the regression model into the system.

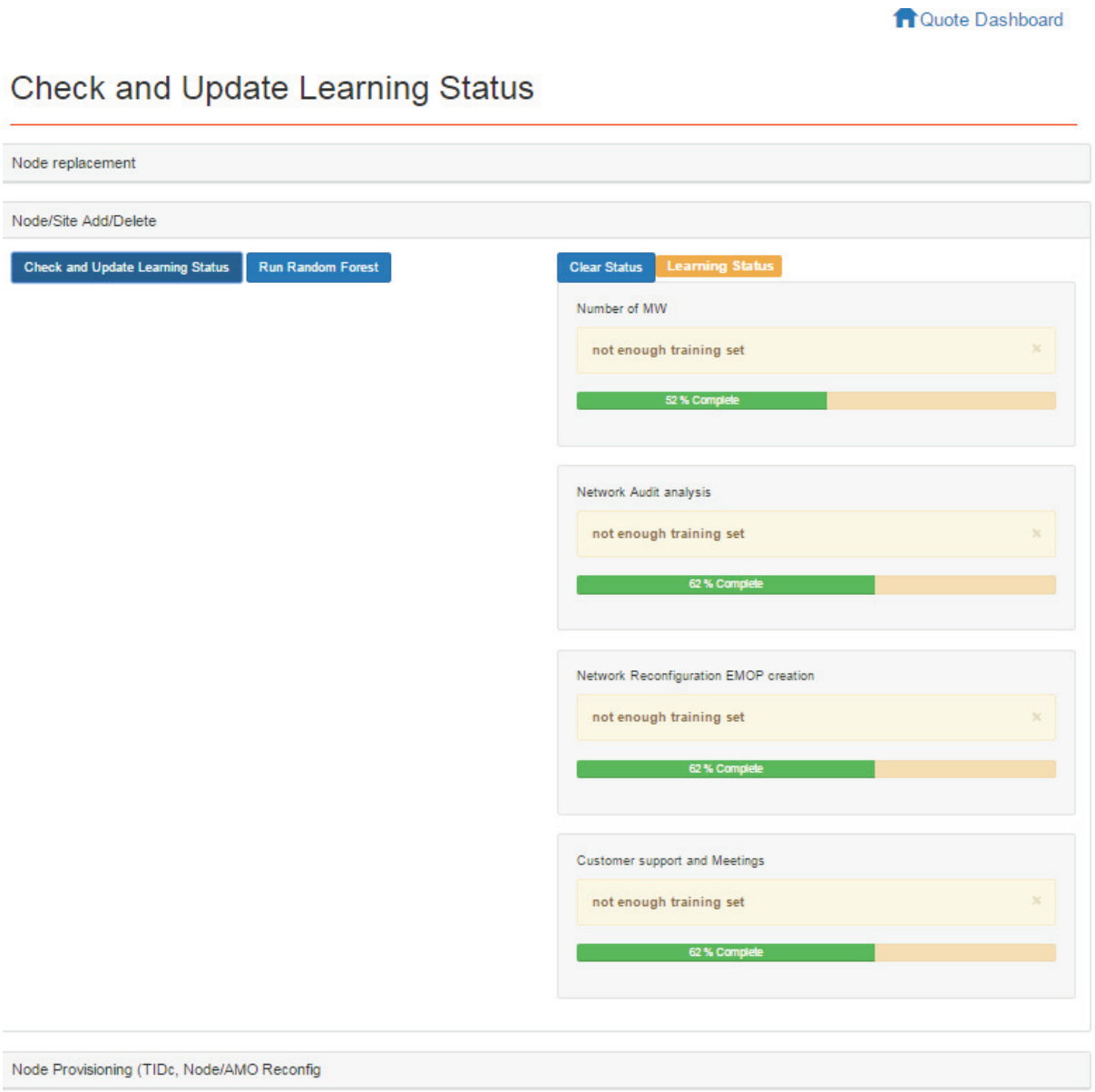

Figure 7: Learning status of output parameters with few data samples

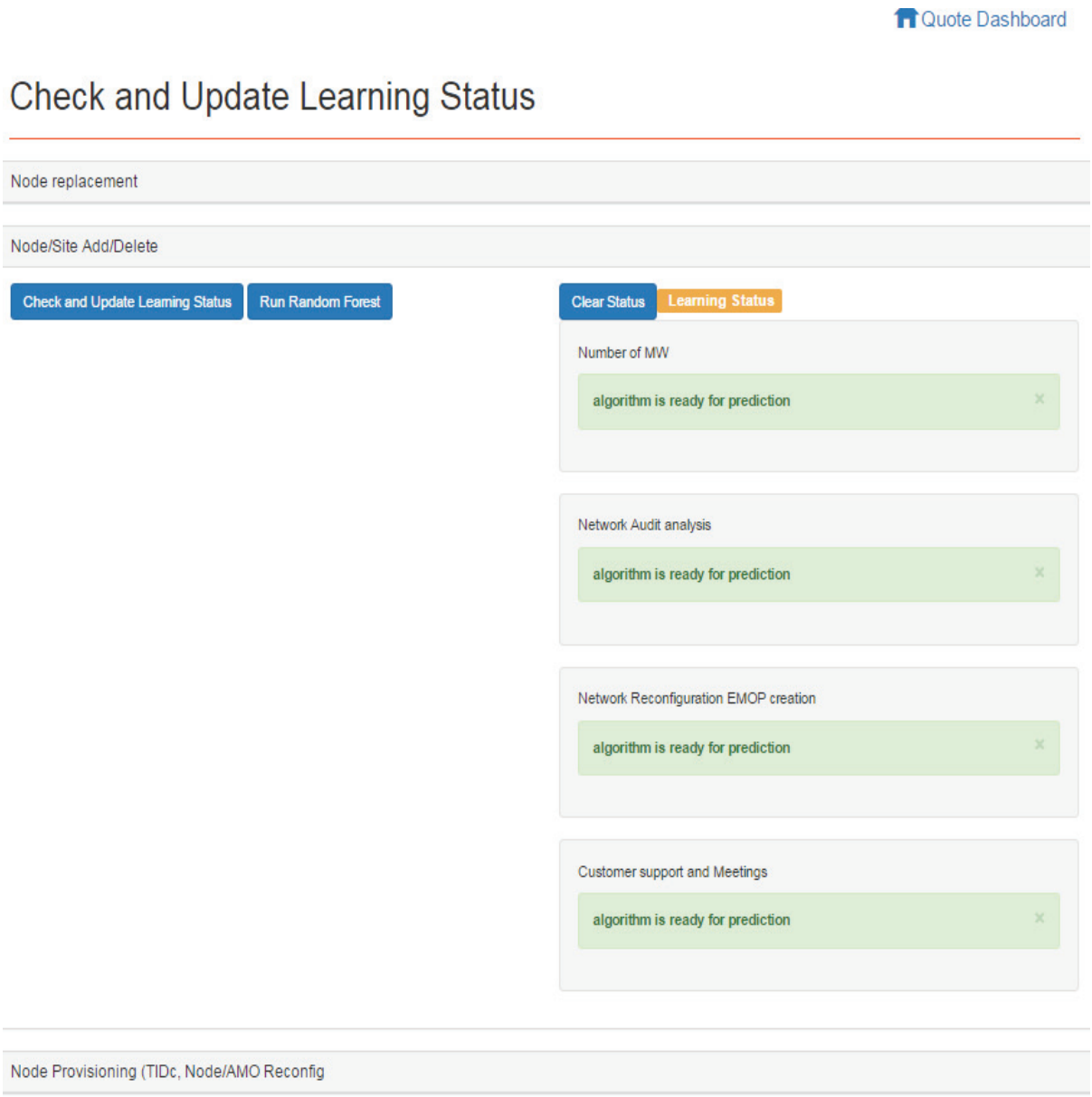

Figure 8: Learning status of output parameters with enough data samples

5.3.2 Selection of Regression Algorithms for Output Parameters

This section shows the chosen regression algorithm for all the output parameters related to the operation named "node/site add/delete". The system has been experimented initially with 25 data samples, then with 32 data samples and finally with 38 data samples. Figures 10 to 12 show the chosen regression algorithm on all the stages mentioned above.

For output parameter named "network audit and analysis", the selected regression after 25 and 32 data samples is Support Vector regression and after 38 data samples, the selected regression is changed to linear Regression.

The elected regression algorithms for the output parameter "network reconfiguration and EMOP creation" is Support Vector regression for 25 data samples and linear regression for 32 and 38 data samples which is shown in the figure 10.

Figure 11 shows selected regression algorithms for output parameter named "customer support and meeting" which are Support Vector Regression, linear regression and linear regression for data samples - 25, 32 and 38 in order.

The chosen regression algorithms for output parameter "number of maintenance windows (MW)" are Support Vector regression, logistic regression and linear regression for data samples - 25, 32 and 38 respectively which is shown in Figure 12.

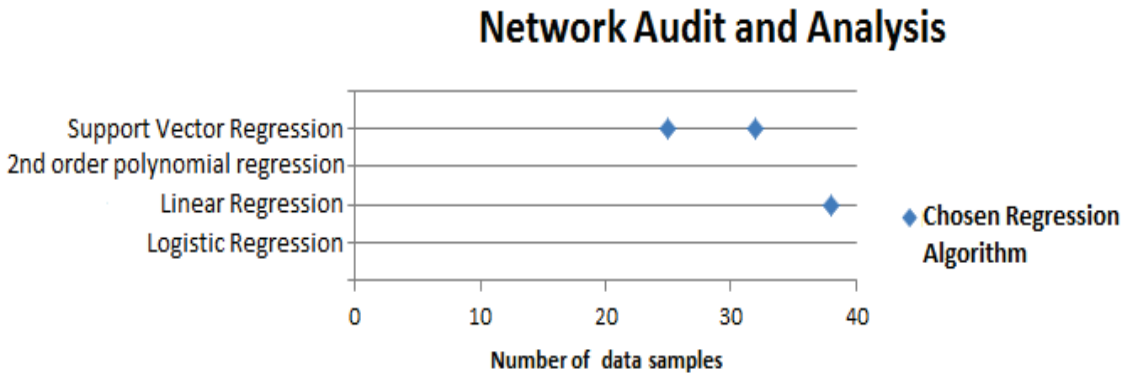

Figure 9: Selected regression algorithm for the output parameter named "Network Audit and Analysis" according to the number of data samples

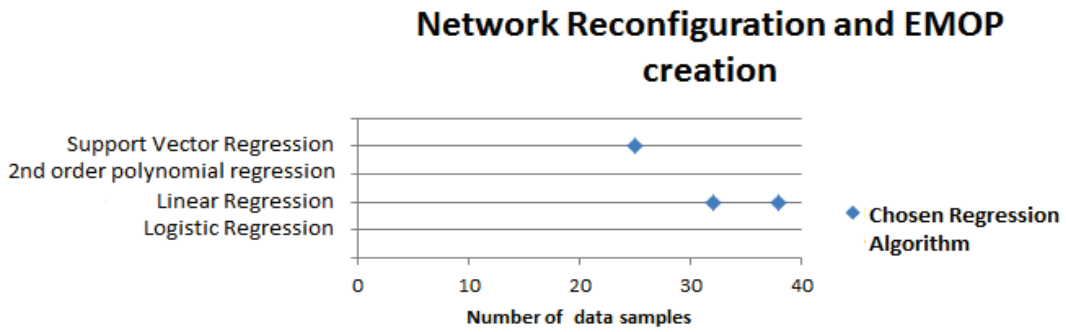

Figure 10: Selected regression algorithm for the output parameter named "Network Reconfiguration and EMOP creation" according to the number of data samples

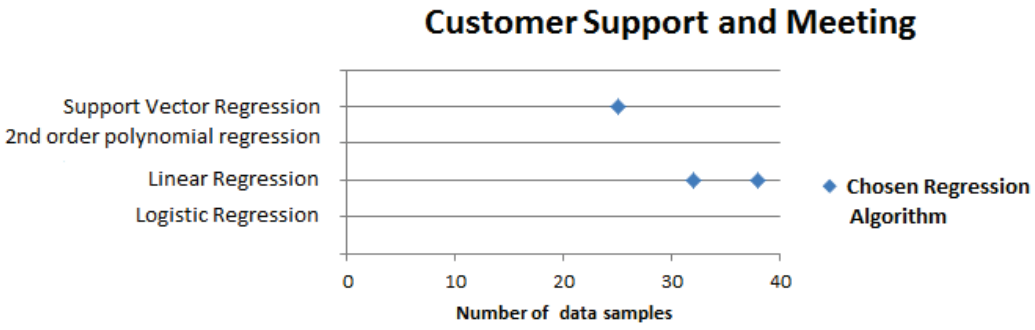

Figure 11: Selected regression algorithm for the output parameter named "Customer Support and Meeting" according to the number of data samples

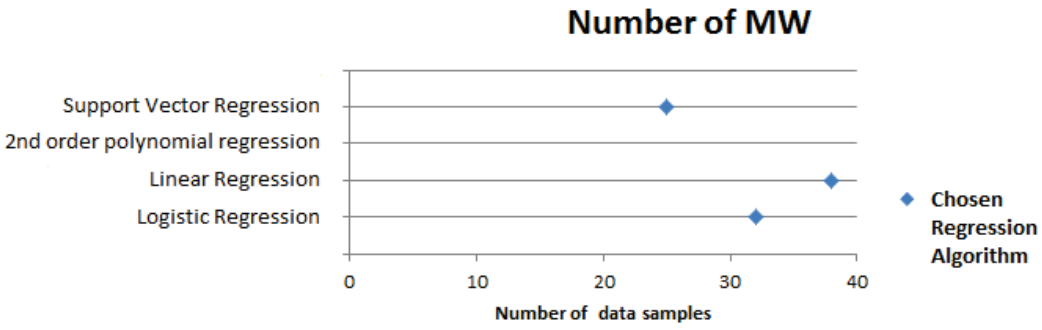

Figure 12: Selected regression algorithm for the output parameter named "Number of MW" according to the number of data samples

5.4 Steady Stage

5.4.1 Prediction Phase

The algorithm is trained, and the regression model is stored in the AQE system. When the network engineer chooses to create a new quote for that operation, the AQE system will feed the value of input parameters into the regression model to get the value for the corresponding output parameter.

The input parameters such as : "number of nodes to add", "number of nodes to delete", "number of sites to add" or delete" and "number of systems involved" and their corresponding values 2, 0, 1, 2, in order, related to the type of operations named "node/Site add/delete" is shown in Figure 13.

After feeding the above values to the corresponding regression model, the value for the output parameters is predicted according to the regression formula. Figure 14 shows that the predicted values for the output parameters - "number of maintenance windows (MW)", "network audit analysis" and "customer support and meeting" are 8, 60 and 1 in an order.

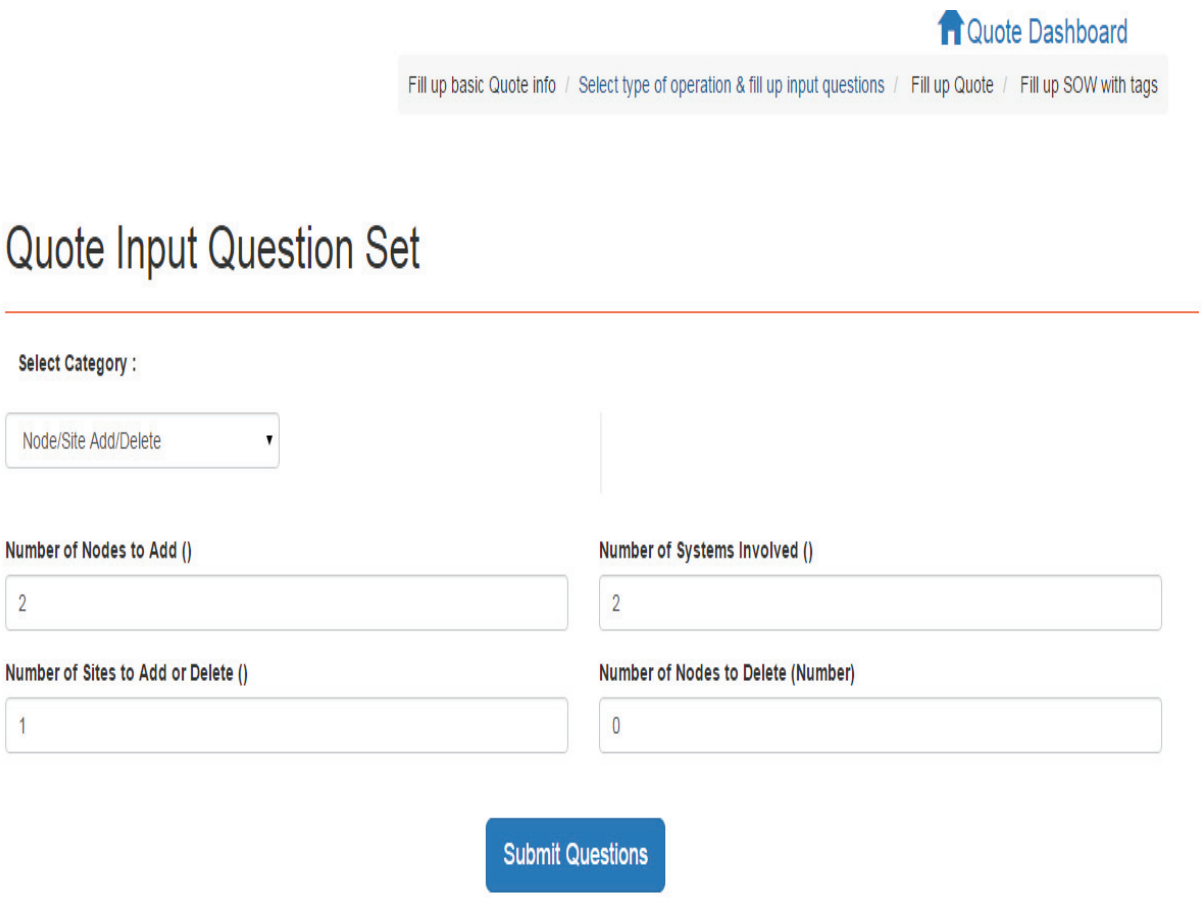

Figure 13: Quote input question set page

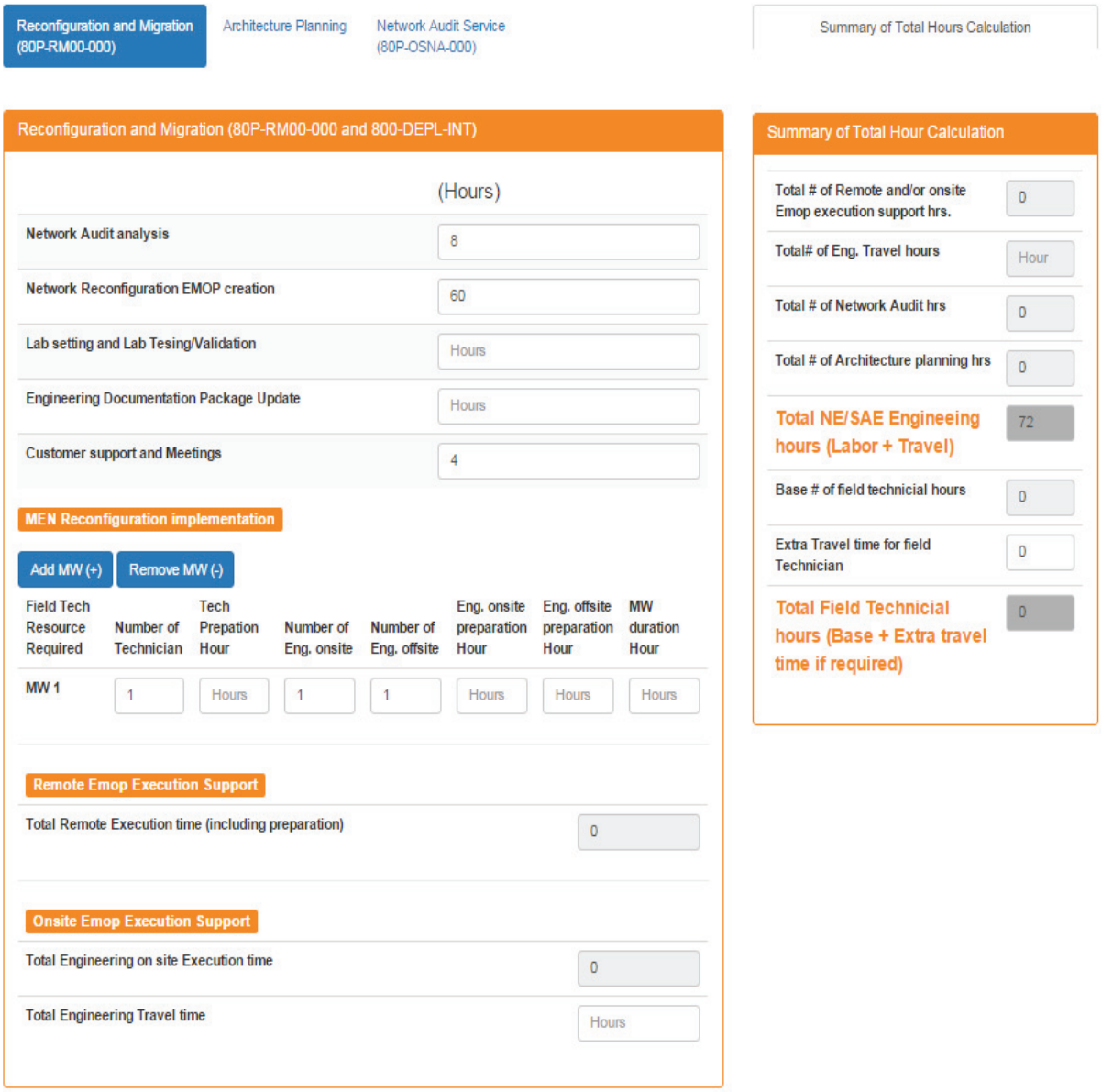

Figure 14: Quote with predicted output parameters value

5.4.2 Feedback Phase

When the AQE system notices a discrepancy between the predicted value and the network engineer's assigned value, is beyond the threshold, it will redirect the user to the feedback module to gather the reason for the discrepancy. As shown in the figure 15, when the network engineer inputs the value - 10 and 55 for input parameters "network audit analysis" and "network reconfiguration EMOP creation", the AQE system knows that the entered value is beyond the threshold error margin and hence, it takes the network engineer to the feedback page.

After accumulating the reasons of discrepancy from network engineer, then the proposed AQE system applies RAKE algorithm on suggestion text to extract the keywords out of it.

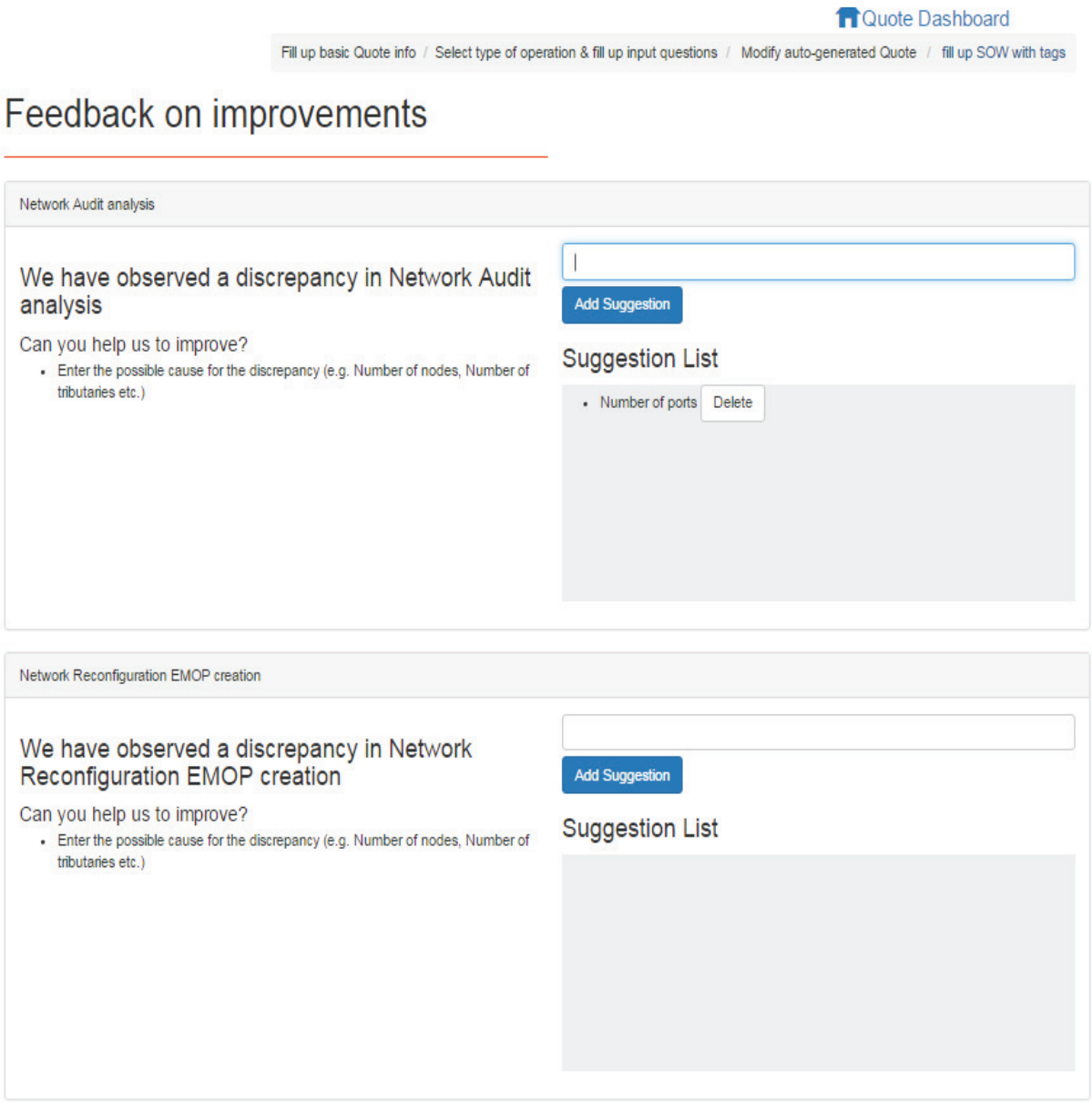

Figure 15: Feedback page for gathering suggestions

5.4.3 Configure a New Input Parameter

The "Configure Input Questions" page is shown in Figure 16 where "Network Size" and "Number of Ports" are two keywords extracted from the network engineer's suggestion for the discrepancy in the output parameter named "network audit analysis". Now, the admin network engineer can format the new question accordingly and can add it to the regression formula to improve prediction accuracy for that output parameter.

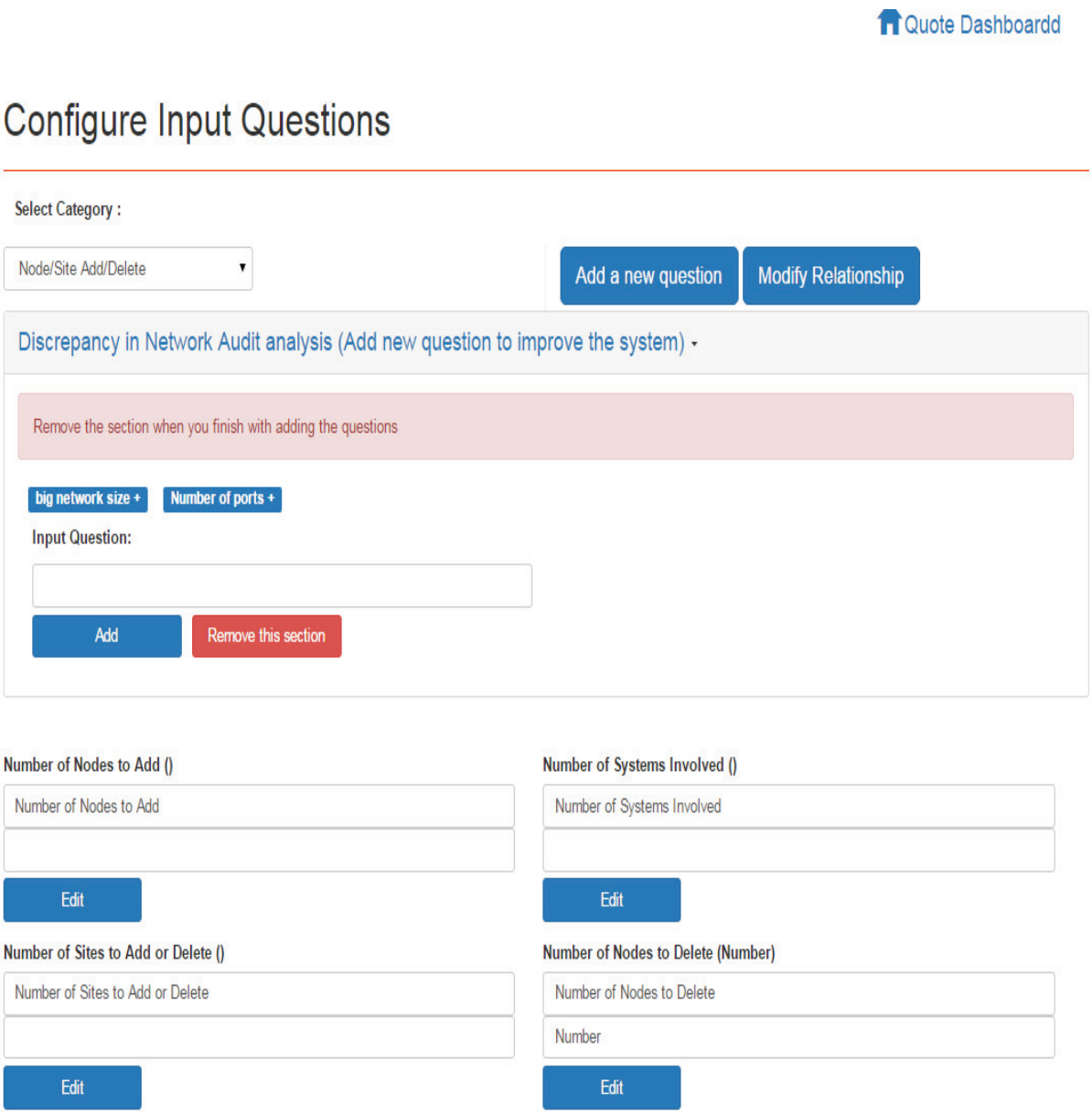

Figure 16: "Configure Input Questions" page

5.4.4 Random Forest for Feature Scoring

As random forest provides good accuracy, robustness and ease of use, we are applying it on the set of output parameters as a feature selection algorithm.

Tables 4, 5, 6 and 7 shows the random forest ranking and scoring for each output parameter - "network audit and analysis", "network reconfiguration and EMOP creation", "customer support and meeting" and "number of MW (maintenance windows)" respectively. These tables have a three columns named "Input parameters", "Score" and "Ranking". Here, "Input parameters" column shows all the input parameters related to particular output parameter, "Score" column displays how much the particular input parameter impacts the regression model performance.

Table 4: Feature scoring for output parameter - "Network Audit and Analysis"

Input Parameters (Features)	Score	Ranking
Total Number of Nodes to Add	0.058	2
Number of Systems Involved	-0.213	3
Number of Site to Add/Delete	0.594	1
Total Number of Nodes to Delete	-4.272	4

Table 5: Feature scoring for output parameter - "Network Reconfiguration and EMOP creation"

Input Parameters (Features)	Score	Ranking
Total Number of Nodes to Add	0.253	2
Number of MW	0.594	1
Number of Systems Involved	-4.272	4
Total Number of Nodes to Delete	-0.508	3

Table 6: Feature scoring for output parameter - "Customer Support and Meeting"

Input Parameters (Features)	Score	Ranking
Network Audit and Analysis	0.211	1
Network Reconfiguration and EMOP creation	0.015	2
Number of MW	-0.089	3

Table 7: Feature scoring for the output parameter - "Number of MW (maintenance windows)"

Input Parameters (Features)	Score	Ranking
Number of Site to Add/Delete	0.227	1
Total Number of Nodes to Delete	0.03	2
Total Number of Nodes to Add	-0.065	3
Number of Systems Involved	-0.203	4

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Fully Automated vs. Semi-automated System

A quotation system with no automation can be time consuming; however, moving to a fully automated system brings limited benefits, and it is not feasible in some cases.

There are few features in the AQE system that need to be fully automated. For instance, the AQE system does not delete irrelevant input questions from the regression formula automatically as it depends on the network engineer to explicitly run the algorithm to remove the input questions. Rather than updating the prediction model by an action of a network engineer, there should be a way for the AQE system to update the model over time based on the frequency of the quote data.

In many cases, it is not feasible to automate all the desired tasks. For instance,

the significance of the output parameter cannot be automatically understood by the AQE system. The AQE system does not have any idea with regards to the margin of error while predicting values. For example, the discrepancy value of "3" between the predicted and the real value for the output parameter - "number of maintenance windows (MW)" is a more serious issue than the same discrepancy value for the output parameter - "customer support and meeting". Hence, to prevent the above mentioned problem, the network engineer sets the threshold value for each output parameter which helps the AQE system to track the activity in the database in case of a discrepancy that goes beyond the threshold limit.

The project was mainly a design of the semi-automated AQE system.

The project was mainly the design of a semi-automated quote system which provides assistance to network engineers to create quotes quickly and efficiently. Therefore, the main focus was provided in implementing the AQE system using machine learning algorithms rather than validation.

The AQE system fulfills this objective by going through three stages. First, in the initialization stage, the network engineer enters the new operation type and builds an association between the input and output parameters. Second, in the early training stage, the AQE system waits for a fair amount of quotes to be gathered and then it builds a prediction model accordingly. Finally, in the steady stage, the AQE system will make predictions and simultaneously gather feedback from the network engineer for prediction improvement.

We have completed implementing the AQE system with all three stages which

satisfy the main objective of the AQE system. According to the number of quotes being entered into the AQE system, it will take at least few years to validate the AQE system properly.

Ciena has underestimated the investment to deploy the AQE system.

The time and effort required to build such quote system were underestimated. Only two network engineers went through all the quotes made in the last year, manually derived a set of categories, and defined questions for each of these categories; this procedure hence took more than enough time. They are not interested in allocating more resources to enter the quotes from the past few years into the AQE system. Lack of participation from network engineers in adding the quotation and providing the necessary feedback to improve the AQE system delayed the development of the product.

Since there were only a small number of quotes in these categories, it was not enough to build a regression model. By observing the number of quotes being entered into the AQE system per year, it would take at least one year to build a regression model which is accurate enough.

6.2 Future Work

Future Steps for Ciena

To check the participation of the input parameters in predicting the specific output parameter, the network engineer has to visit the learning controller and run the

random forest regressor explicitly. Instead, we are thinking of notifying the network engineer automatically whenever some input parameters show poor participation in output parameter prediction.

Building and updating the regression models for prediction is not fully automated yet. After accumulating sufficient number of quotes, the network engineer has to run the decision library to build or to update the regression model every time. Instead, we can plan the AQE system to trigger the module to refresh the regression model over the time automatically.

There is a possibility that the same questions might be added into the regression formula with different wording which does not contribute much to the prediction of the associated output parameter. Therefore, we are thinking about checking for duplicate questions before adding it into the regression formula.

Future Steps for Research

In the feedback module, we allow the network engineer to provide feedback in plain text. Thereafter, we apply the keyword extraction algorithm for extracting the keywords. From the extracted keywords, the admin network engineer builds few questions which can possibly be included in the regression formula. He/she distributes these questions to a group of network engineers who vote on these questions. The highest voted question, assumed to be relevant enough would be added to the regression formula.

The next step in terms of the research is to make the AQE system more generalized so that it can be applicable to other domains as well. For example, house price

prediction. In [42], the author uses multiple linear regression algorithms for building a regression model for prediction. Our AQE system applies three other regression algorithms on top of the multiple linear regression algorithm for building a regression model, and can also use the feedback module from our AQE system to improve the prediction.

Finally, after gathering a large number of quote data, the validation of the AQE system needs to be performed to assess its accuracy.

Bibliography

[1] Scikit-learn, Machine Learning in Python. http://scikit-learn.org/. Last access: 28 August 2015.

[2] Apache Lucene. http://lucene.apache.org. Last access: 28 August 2015.

[3] Axiom Sales Manager. http://www.raeko.com/. Last access: 4 November 2015.

[4] Selecting Good Features. http://goo.gl/FniVWl. Last access: 28 August 2015.

[5] Machine Learning. http://whatis.techtarget.com/definition/machine-learning. Last access: February 2016.

[6] How To Find Relationship Between Variables, Multiple Regression. http://www.statsoft.com/textbook/multiple-regression. Last access: 2016-04-16.

[7] A Python Implementation of the Rapid Automatic Keyword Extraction. https://github.com/aneesha/RAKE. Last access: 28 August 2015.

[8] Quote Werks. http://www.quotewerks.com/. Last access: 4 November 2015.

[9] Support Vector Machine Regression. http://kernelsvm.tripod.com/. Accessed: 2016-04-16.

[10] Workflow Max. http://www.workflowmax.com/. Last access: 4 November 2015.

[11] M. Anitescua, O. Rodericka, P. Fischera, and WS. Yangb. Polynomial Regression with Derivative Information in Nuclear Reactor Uncertainty Quantification. 2009.

[12] Apptus. http://www.apptus.com. Last access: 4 November 2015.

[13] Apttus Using Machine Learning in Price-Quoting Management. http://www.eweek.com/enterprise-apps/apttus-using-machine-learning-in-price-quoting-management.html. Last access: 25 September 2015.

[14] A. Balahur. Sentiment Analysis in Social Media Texts. In *4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 120–128, 2013.

[15] J. Benesty, J. Chen, Y. Huang, and I. Cohen. Pearson Correlation Coefficient. In *Noise Reduction in Speech Processing*, pages 1–4. Springer, 2009.

[16] M. W. Berry and J. Kogan. *Text Mining: Applications and Theory*. John Wiley & Sons, 2010.

[17] S. Biondo, E. Ramos, M. Deiros, J. M. Ragué, J. De Oca, P. Moreno, L. Farran, and E. Jaurrieta. Prognostic Factors for Mortality in Left Colonic Peritonitis: a New Scoring System. *Journal of the American College of Surgeons*, 191(6): 635–642, 2000.

[18] C. R. Boyd, M. A. Tolson, and W. S. Copes. Evaluating trauma care: The triss method. *Journal of Trauma and Acute Care Surgery*, 27(4):370–378, 1987.

[19] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[20] J. Bruin. Newtest: Command to Compute New Test @ONLINE, February 2011. URL `http://www.ats.ucla.edu/stat/stata/ado/analysis/`.

[21] C. Cortes and V. Vapnik. Support vector machine. *Machine Learning*, 20(3): 273–297, 1995.

[22] F. d Alche-Buc and L. Ralaivola. Incremental Learning Algorithms for Classification and Regression: Local Strategies. In *AIP Conference Proceedings*, pages 320–329, 2002.

[23] D. A. Freedman. *Statistical Models: Theory and Practice*. Cambridge University Press, 2009.

[24] K. J. Friston, A. P. Holmes, K. J. Worsley, JP. Poline, C. D. Frith, R. SJ.

Frackowiak, et al. Statistical Parametric Maps in Functional Imaging: a General Linear Approach. *Human Brain Mapping*, 2(4):189–210, 1994.

[25] B. P. Green and J. H. Choi. Assessing the Risk of Management Fraud through Neural Network Technology. *Auditing*, 16(1):14, 1997.

[26] L. Huang, J. Jia, B. Yu, B. Chun, P. Maniatis, and M. Naik. Predicting Execution Time of Computer Programs Using Sparse Polynomial Regression. In *Advances in Neural Information Processing Systems*, pages 883–891, 2010.

[27] A. Hulth. Improved Automatic Keyword Extraction Given More Linguistic knowledge. In *Conference on Empirical Methods in Natural Language Processing*, pages 216–223, 2003.

[28] M. Kevin. *Machine Learning: a Probabilistic Perspective*. The MIT press, 2012.

[29] J. Kincaid. EdgeRank: The Secret Sauce That Makes Facebook's News Feed Tick. *TechCrunch, April*, 2010.

[30] M. Kologlu, D. Elker, H. Altun, and I. Sayek. Validation of MPI and PIA II in Two Different Groups of Patients with Secondary Peritonitis. *Hepato-gastroenterology*, 48(37):147–151, 2000.

[31] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis. Machine Learning Applications in Cancer Prognosis and Prediction. *Computational and Structural Biotechnology Journal*, 13:8–17, 2015.

[32] R. D. Lawrence. A Machine-Learning Approach to Optimal Bid Pricing. In *Computational Modeling and Problem Solving in the Networked World*, pages 97–118. Springer, 2003.

[33] J. Le Gall, S. Lemeshow, and F. Saulnier. A New Simplified Acute Physiology Score Based on a European/North American Multicenter Study. *Jama*, 270(24): 2957–2963, 1993.

[34] J. Leskovec, A. Rajaraman, and Jeffrey D. Ullman. *Mining of Massive Datasets*. Cambridge University Press, 2014.

[35] A. Liaw and M. Wiener. Classification and Regression by Random Forest. *R News*, 2(3):18–22, 2002.

[36] J. C. Marshall, D. J. Cook, N. V. Christou, G. R. Bernard, C. L. Sprung, and W. J. Sibbald. Multiple Organ Dysfunction Score: a Reliable Descriptor of a Complex Clinical Outcome. *Critical Care Medicine*, 23(10):1638–1652, 1995.

[37] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, et al. Ad Click Prediction: a View From The Trenches.

[38] R. Mihalcea and P. Tarau. TextRank: Bringing Order into Texts. Association for Computational Linguistics, 2004.

[39] L. Nunno. Stock Market Price Prediction Using Linear and Polynomial Regression Models. Technical report, University of New Mexico, 2014.

[40] S. K. Palei and S. K. Das. Logistic Regression Model for Prediction of Roof Fall Risks in Bord and Pillar Workings in Coal Mines: an Approach. *Safety Science*, 47(1):88–96, 2009.

[41] Jeff Palmer and Arijit Chakravarty. Supervised machine learning. *An Introduction To High Content Screening: Imaging Technology, Assay Development, and Data Analysis in Biology and Drug Discovery*, page 231, 2014.

[42] I. Pardoe. Modeling Home Prices Using Realtor Data. *Journal of Statistics Education*, 16(2), 2008.

[43] Quotient. `https://www.quotientapp.com/`. Last access: 4 November 2015.

[44] J. Ramos. Using TF-IDF to Determine Word Relevance in Document Queries. In *Proceedings of the First Instructional Conference on Machine Learning*, 2003.

[45] H. L. Seal. The Historical Development of the Gauss Linear Model. *Kendall and Pearson, Studies*, 1968.

[46] S. Sen, S. K. Lam, Al M. Rashid, D. Cosley, D. Frankowski, J. Osterhouse, F. M. Harper, and J. Riedl. Tagging, Communities, Vocabulary, Evolution. In *20th Anniversary Conference on Computer Supported Cooperative Work*, pages 181–190, 2006.

[47] S. Sen, J. Vig, and J. Riedl. Tagommenders: Connecting Users to Items Through

Tags. In *18th International Conference on World Wide Web*, pages 671–680, 2009.

[48] G. Smith. *Tagging: People-Powered Metadata for the Social Web, Safari*. New Riders, 2007.

[49] A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. *Statistics and Computing*, 14(3):199–222, 2004.

[50] K. Sparck Jones. A Statistical Interpretation of Term Specificity and Its Application in Retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

[51] M. Strano and B. M. Colosimo. Logistic Regression Analysis for Experimental Determination of Forming Limit Diagrams. *International Journal of Machine Tools and Manufacture*, 46(6):673–682, 2006.

[52] J. Truett, J. Cornfield, and W. Kannel. A Multivariate Analysis of the Risk of Coronary Heart Disease in Framingham. *Journal of Chronic Diseases*, 20(7): 511–524, 1967.

[53] X. Yan. *Linear Regression Analysis: Theory and Computing*. World Scientific, 2009.