10-8-2018

A Decision Model for Data Mining Techniques

Maoloud Dabab
Portland State University

Mike Freiling
Conceptrics AG

Nayem Rahman
Portland State University

Daniel Sagalowicz
Portland State University

Citation Details

M. Dabab, M. Freiling, N. Rahman and D. Sagalowicz, "A Decision Model for Data Mining Techniques," 2018 Portland International Conference on Management of Engineering and Technology (PICMET), Honolulu, HI, 2018, pp. 1-8.

A Decision Model for Data Mining Techniques

Maoloud Dabab[1], Mike Freiling[2], Nayem Rahman[1], Daniel Sagalowicz[1]
[1] Engineering & Technology Management Dept., Portland State University, Portland, Oregon USA
[2] Conceptrics AG, Portland, Oregon USA & Kyoto, Japan

***Abstract*—Data mining is the process of extracting useful information from very large data sources. Data mining techniques have proven to be very useful in many domains. However, there is no single algorithm or technique that works best across all types of datasets and problems, and it remains "an art" to decide what data mining technique to use for a specific situation. This paper surveys several data mining techniques that can be applied to different business problems, and presents a decision model in the form of a series of 15 – 20 questions that help identify the best approach or approaches to a specific problem at hand. For some sets of answers, a small number of techniques are dominant. The decision model is based on a review of the current literature, as well as expert experience. The fraud detection problem is adopted as a case study and applied the data mining techniques to draw the insights. We also discuss the applicability of specific techniques to common business in finance, marketing, and business operations.**

I. Introduction – What's the Problem?

The desire to use available data effectively has permeated most commercial entities of any size, as well as numerous government agencies and non-profit organizations. The sudden upsurge of interest, of course, has led to a shortage of skilled workers, leaving many organizations with the twin challenges of trying to make sense of the available technologies on their own, and learning how to utilize them to make effective contributions to organizational goals.

These challenges are exacerbated by the bewildering array of algorithms available, from a dauntingly divers set of different technical approaches. The fledgling "big data" impresario, then must answer a host of critical questions, often without adequate resources to develop thoughtful and comprehensive answers. Such questions include:

- What algorithms and technologies should I invest time and effort in?
- What data should I apply to these algorithms?
- What constraints on the data must I observe?
- How do I know when to abandon investment in one technique for another?
- How do I know when I'm done investigating, and can proceed to do "real" work?

The purpose of this paper is to lay out a framework that will enable such organizations to answer at least some of these questions. Starting with a representative but diverse sample of available techniques, we identify key characterizes, and key constraints on the data required. We demonstrate how these characteristics can be used to organize a decision process that helps in selecting initial candidates for exploration, and we validate this approach to decision making with some empirical results based on a single data set. Finally, we discuss how this part of the decision making process fits into the overall "analytics architecture" and "analytics life cycle" that any organization must develop in order to utilize data analytics effectively.

II. Prior Work

Now-a-days, most business organizations hold huge volumes of data. These data consist of internal transactional data and external business related data, which get generated in many places including the social media sites. These data are stored in traditional data warehouses as well as Hadoop-based big data platforms [24, 7]. Due to enormous volume of data knowledge workers need special tools, technologies and data mining algorithms or techniques to process, perform predictive analytics, and analyze the data. It requires processing of data using extract-transform-load tools, business intelligence tools [25, 23] and data mining tools [15, 24] in order to make informed business decisions.

Survey of literature on data mining techniques and applications from 2000 to 2018 [1, 12, 28, 36] reveals that data mining has been gaining popularity and its applications appear to work as driving forces to provide new understanding in many real world problems including business, economic, social, and psychological. There are several specific business areas in which data mining techniques are extensively used: fraud detection [1, 21], bankruptcy prediction [20, 27, 3, 31, 18], customer relationship management [26], and customer profiling to name a few. Data mining techniques are widely used to detect credit card and other financial fraud detection and found to be effective [5, 30, 4]. Each year billions of dollars are lost because of a varieties of financial frauds. Data mining techniques are used to build predictive models and provide decision makers with valuable information. Data mining techniques are capable to show surprising behavior of hidden data, or detect unknown malicious attacks more accurately [2].

To solve real world problems, researchers in industry and academia have come up with different data mining techniques. But there is no single source that provides which technique solves what problem and how, advantages and disadvantages, use-case examples. Recently, researchers have come up with top ten data mining algorithms that cover different data mining techniques. They include C4.5, k-Means, SVM, Apriori, EM, PageRank, AdaBoost, kNN, Naïve Bayes, and CART [11, 32] . Prominent data mining techniques include decision trees, support vector machine, neural networks, k-nearest neighbors,

k-means, Bayesian networks, and logistic regression [32] which are driven by top 10 algorithms mentioned above.

In data mining world, classification is considered as one of the data mining problems as opposed to data mining technique [36]. There are many data mining techniques proposed to solve classification problems: decision trees [22, 16], support vector machine [29], k-nearest neighbors [11, 22], logistic regression [33], Bayesian networks [17, 22], neural networks [10, 14]. For anomaly detection quite a few data mining techniques are used including Naïve Bayes network, neural networks, and support vector machine [2]. The data mining techniques that solve classification problems fall under supervised learning. Phyu [22] observes that for classification there is no conclusion as to which data mining technique is the best.

Decision trees provide predictive analytics based on certain rules using if-then-else. It is a graphical representation of relationships among variables in tree like format. For bankruptcy prediction decision trees were found to be more accurate compared to neural networks and support vector machines [20]. Gepp et al. apply decision trees models to predict business failures [8]. Decision trees is considered as one of the most popular data mining and presentation techniques [5]. One limitation of decision trees is that it generates too many rules [20]. Decision Trees is easy to interpret and understand but, it generates too many rules.

The k-means under clustering algorithm is suggested as one of the techniques for the anomaly detection which is able to do intrusion detection without prior knowledge [2]. Likas et al. apply the global k-means algorithm to solve data partitioning problem [13]. The k-means generates a number of groups from a given dataset to put the identical values or transactions under some predefined clusters, k. User can define number of clusters and k-means returns results accordingly. The k-means is seen as a popular clustering technique for data mining [11, 9].

The support vector machine (SVM) consists of supervised learning techniques that are used for classification and regression. The SMV classification separates the target classes and on the other hand, the SMV regression builds continuous function with data points [6]. This algorithm is insensitive to the number of dimensions and requires only a few examples for training [32]. One-class SMV is used to build profile for anomaly detection [6]. The SVM is considered as one of the most robust and accurate algorithms. Agrawal and Agrawal [2] report that SMV outperforms in terms of accuracy when compared with neural-network techniques. Shin et al. attempt to solve bankruptcy prediction problem by comparing support vector machines and neural network [27]. They report that support vector machines perform better than neural network in terms of accuracy and performance.

Neural network consists of a bunch of nodes each of which has weighted connection to other nodes. Ogwueleka uses neural network technique for the credit card fraud detection by apply a technique that put transaction data into four clusters of low, high, risky and high-risk clusters [19]. This method was able to detect over 95% of fraud cases. Odom and Sharda [18] propose a neural network model for bankruptcy prediction. Their model was tested with financial data and concluded that neural network model is capable to solve bankruptcy prediction problem. Atiya [3] develops a neural network model for bankruptcy prediction for credit risk. The author proposes some indicators for the neural network system which is able to improve prediction accuracy from 81.46% (based on traditional credit risk models) to 85.5%.

The k-nearest neighbor algorithm takes entire training data into memory to perform classification [32]. When new un-labeled data comes in, kNN uses distance metrics to compare with k-nearest neighbors in training data set and then makes decision to classify it [11]. There two issues with the performance of k-NN regarding choice of k. A too small k can cause results to be sensitive to noise points and a too large k the neighborhood might include too many points from other classes [32].

III. Our Methodology

The task of developing effective decision models, architectures, and life cycles for data analytics in the enterprise is both broad and complex. Consequently, it must be tackled in stages. The first stage, of course, is to sketch the outlines of an initial framework that captures the essential elements while remaining open to adaptation and refinement based on further research and actual experience.

That is the goal of this paper – to outline a framework that supports the preliminary efforts of an organization to make early decisions (prior to, and thus in the absence of, actual experience). In designing this preliminary framework, we have relied on qualitative interviews with experienced practitioners and personal observation of the capabilities of specific techniques.

While these preliminary concepts have not been validated in any quantitative, scientific fashion, we believe they are highly suggestive of the fruitfulness of such a framework, as well as of promising direction for future research. We offer these preliminary results in the spirit of "opening a dialog" on this important subject, and we welcome other contributions.

In order to make progress, we have limited the scope of the current investigation to techniques for addressing problems involving classification and/or prediction – the two are closely related. We have chosen not to focus on techniques for association and/or recommendation problems (e.g. "Here is a book you might like"), although we make passing reference to techniques that are also well suited to these types of problems.

In an effort gain some preliminary validation of these concepts, and to clarify our exposition, we develop a case study from a specific business problem (fraud detection) where one of the authors has significant experience. Our hope is that this case study will serve as an example that enables others to work out similar results for their own problem domains.

IV. Analytical Techniques

The decision model we developed encompasses 9 different modeling approaches. Here are the approaches, with a summary of each:

A. *Neural networks (NN)*

Take linear combinations of the inputs and then make nonlinear transformations of the linear combinations using an activation function.

B. *K-Means (KM)*

Find K points that are considered as potential means of the clusters. Find where the true means are for those K clusters. Redo.

C. *Regression (REG)*

Regression analysis is a statistical technique for investigating and modeling the relationship between variables. In other words, it is a statistical tool for evaluating the relationship of one or more independent variables to a continuous dependent variable. In regression analysis, independent variables are classically continuous, but in practice, any type can be used.

D. *Classification and Regression Tree (CART)*

CART is a method that refers to Decision Tree algorithms that can be used for classification or regression predictive modeling problems. This was built for predicting continuous dependent variables and categorical predictor variables, which are obtained by recursively partitioning the data space and fitting a simple prediction model within each partition.

E. *Decision Trees (DT)*

Is a tree-like way to represent a collection of hierarchical rules that leads to a class or value. This technique can be applied to solve classification tasks only.

F. *K-Nearest Neighbors (KNN).*

Find the K samples that are closest to the case being considered. And assume majority is the true choice.

G. *Support Vector Machines (SVM)*

The SVM is a training algorithm for learning classification and regression rules from data.

H. *Scorecard Autotuning (SCA)*

Known decision variables are assigned an initial set of scoring schedules, which are adjusted via constrained optimization.

I. *Random Forest (RF)*

Random forests is an ensemble learning method that creates a number of trees, all slightly different, as a forest, which each tree depends on the values of a random vector sampled independently, and the result is a combination of tree predictor for all trees in the forest.

V. Selection Criteria

The following selection criteria were adopted, based on interviews with experiences practitioners:

- Supervised vs. unsupervised
- Number of observations
- Number of independent variables
- Type of independent variables
- Quality of independent variable values
- Missing values
- Standardization of values
- Number of dependent variables
- Density of dependent variables
- A priori knowledge of dependent variable structure
- Risk of overfitting
- Availability of second order metrics
- Support for explanation / justification / human review

VI. The Decision Model

The complete decision matrix is shown in Appendix A below. In this section, we analyze the spreadsheet to come up with a decision process to decide which technique(s) to try for specific business situations. First, we are going to divide business situations into two categories: classification and prediction.

A. *Problem category is classification*

If Problem category is classification, the first differentiation we will consider is whether we are in a supervised or unsupervised situation.

a) Problem category is classification/unsupervised

At this point, let's look at the number of independent variables. From the spreadsheet, if number of independent variables is

- Large → use Neural Networks
- Medium → use K-Means

b) Problem category is classification/supervised:

- From the spreadsheet, if number of observations is Sparse → use Support Vector Machines.
- Otherwise if number of observations is medium: let's look at the number of independent variables. If number of independent variables is:
  - Medium → use K-Nearest Neighbors
  - Large → use Scorecard Auto tuning
- Finally, if number of observations is large, again let's look at the number of independent variables.
  - If number of independent variables is small → use Regression
  - Otherwise if number of independent variables is large
    - if you need explanation
      → use CART, or
      → Random Forest
    - If you do not need explanation
      → Neural Networks may ALSO be a good choice
    - Finally, if the problem is non-linear
      → use Random Forest, or
      → Neural Networks

B. Problem category is prediction

If Problem category is prediction, we are almost always in a supervised situation. Then

a) If number of independent variables is small
   - → use Regression

b) Otherwise (supervised and number of independent variables is large)
   - → use CART, or
   - → Decision Trees

To illustrate the decision model in a concrete setting, we apply the decision model created above to a specific task – fraud detection. The problem category is classification, and learning must be supervised. The number of observations is large, and the number of independent variables is medium. Finally, an explanation is important. Our decision model recommends CART or Random Forest. Next, let's see what the actual test results tell us.

VII. Testing The Decision Model

In this section, we apply a selected set of our modeling techniques to a single task (fraud detection) based on common data sets. Interestingly, these test results favor the same methods, namely CART and Random Forest, as the decision model we developed and applied above.

A. Data Sets Used for Testing

a) The BankSim dataset

This data is an agent-based simulator of bank payments based on a sample of aggregated transactional data provided by a bank in Spain. The preparation of this dataset was done in 2014 [34]. This data is public data, which does not include the customers' information and exposure to legal and private customers' transaction. The data contains around half a million of records of transactional data of six months overlay, restricted from November 2012 to April 2013. The dataset has basic statistical information about the customers' payments that analysis on one of the biggest zip code, and the information are:

- Customer ID was provided which has nothing to do with the reality of customer's information.
- Age Categories are given regarding six groups and the last one as unknown age.
- Gender categories are given as enterprise, female, male, and unknown
- The merchant ID was specified
- There are 16 differentiate payments made which indicted as merchant categories.
- All the prices of the payments given are in euro
- Finally, the fraud situation was explained either 1 or 0

b) The Creditcard Dataset

As a part of validation, we used another dataset to implement the same techniques. The second dataset also public data and was collected and analyzed by Machine Learning Group [35]. It has around two hundred thousand transactions for two days that made by credit cards in back to 2013 by European cardholders. The frauds account for 0.172% of all transactions, and it contains thirty numerical input variables and fraud case output. This dataset confidential so the original features and more background information about the data were hidden. Fig 3 shows that there are no missing values in this data also. Thus, the results were added to the table I.

B. Testing Methodology

The goal of this section to see which of the nine data mining techniques work better with fraud detection dataset as well as the challenges for each method. We used R environment, a statistical computing language, as an appliance to analysis the data using different techniques. R is an open-source statistical software and application of programing language, which enable people to write their packages. The appendix B shows all the libraries and functions that are used in this implementation. We organized the work by dividing our work on R to four steps, which are:

- Uploading and cleaning the data:

First, we upload the data to R environment, and we show all variables to make sure they are in numeral, and they are not characteristics, comma, and other staff. Also, it is very important to choose the right predictors, and this often depends on the subject area and goals of your study. Therefore, from our experience we chose the related predictors in this case. The Table I shows a sample of first ten rows with the header of BankSim dataset.

TABLE I. BankSim Dataset Sample

Customer	Age	Gender	Gender_new	Merchant	Category	Category_new	Amount	Fraud
1093826151	4	M	1	348934600	es_transportation	13	4.55	0
352968107	2	M	1	348934600	es_transportation	13	39.68	0
2054744914	4	F	2	1823072687	es_transportation	13	26.89	0
1760612790	3	M	1	348934600	es_transportation	13	17.25	0
757503768	5	M	1	348934600	es_transportation	13	35.72	0
1315400589	3	F	2	348934600	es_transportation	13	25.81	0
765155274	1	F	2	348934600	es_transportation	13	9.1	0
202531238	4	F	2	348934600	es_transportation	13	21.17	0
105845174	3	M	1	348934600	es_transportation	13	32.4	0

TABLE II. RESULTS OF THE IMPLAMENTING

NO.	Data Mining Technique	MSE (BankSim)	MSE (Creditcard)	Note
1.	Random Forest	0.00334	0.00036	You can decide how many trees, and it seems is the best one
2.	Classification and Regression Tree	0.00493	0.00031	The functions provide many details
3.	Regression	0.00869	0.00062	You can get the same result in multiple ways
4.	K-Nearest Neighbors	0.00970	0.00083	Depends on how much the k value, but so far this is the best results
5.	K-Means	0.01235	0.00180	Very simple and easy to implement
6.	Support Vector Machines	0.01976	0.00235	It takes a long time to get the results "This is a sample of the dataset that we have."
7.	Neural Networks	0.03591	0.00431	In the rustle, you can get the picture of the network and how many steps
8.	Decision Trees	0.05622	0.01340	Very simple and easy to implement
9.	Scorecard Autotuning	---	---	We cannot implement this technique with R

- Checking the quality of the data regarding missing data:

There is a function in R is investigating the missing value if the data has some, but in this data, we do not have missing values for both.

- Analyzing the data for each technique to get the results: primarily in this step the data was divided into groups, random sample of 75% records are taken as training dataset and 25% testing dataset. Then, we use different packages and training dataset to build a model for each technique. After that, we predicate the fraud using the building model and calculate the mean square error for each data separately.
- Building the comparison between all the techniques: The table II summarizes the results of the implementation and gives the challenges and issues for each technique.

C. Results of Testing

To evaluate these test results, we use the root mean square error, of each technique. Using this criterion, we find that the order of performance is roughly the same for both datasets. Notably, the top performing techniques are CART and Random Forest, the same two techniques that were recommended by our decision model, which was developed independently of the testing process. Random Forest has the best performance on the BankSim data, while CART has the best performance on the Creditcard data set.

Following the two top performers, Regression and K-Nearest Neighbors also show reasonably good results. Decision Trees, K-Means, Support Vector Machines, and Neural Networks show relatively poorer results. Scorecard Auto tuning was not evaluated because this is a highly idiosyncratic technique for which a general R package is not available.

It is also worth noting that the mean square error with the Creditcard dataset less that BankSim dataset for all techniques, which means we came with a better model for fraud with Creditcard dataset. The larger number of independent variables in the Creditcard dataset (29) compared to BankSim (6) is likely a major factor. Note, however, that the five best techniques on BankSim outperform the weakest technique (Decision Trees) on the Creditcard data set.

VIII. THE DECISION MODEL IN PRACTICE

A. Analytics Architecture

To be effective, any model must be embedded in an architecture that supports integration of the analytics capabilities with other system elements that are required for performing the task at hand. Here we identify some of the key elements that are typically required in an analytics architecture.

B. Judgmental rules.

Judgmental rules (often referred to a policy rules) surround and support the conclusions provided by the model. These judgmental rules typically enforce limit and "safety net" conditions. In the case of fraud detection, for example, it is always advisable to have a safety net rule that alerts on transactions over some extreme limit for example "Alert on any transaction over $1M in amount".

In addition to safety net rules, judgmental rules may also be needed to enforce non-model constraints required by the ethical, legal, and regulatory context in which the task is performed. An example of such a rule for medical treatments might be "If the treatment recommended is experimental, the family must also be consulted."

Still other judgmental rules are often needed to implement a firm's service philosophy, for example, "If the transaction amount in dispute is less than $25.00 and the customer is a good one, allow the transaction to proceed."

C. Decision pipeline.

Once the analytical models and judgmental rules have been applied to a particular case, the system has done what it can. There is typically a recommendation, and often supporting reasons for the recommendation. Before a final decision can be made, and action taken, there is often a pipeline or workflow to be followed. In some cases (e.g. credit card transactions, purchase recommendations), real time response is required, so the pipeline is very short – direct to the system that executes the credit card decision, or displays the recommendation to a customer.

In many other decision tasks, however, the pipeline must be longer. Human review is often required for difficult or high-risk decisions. In some cases, key stakeholders (customers, patients,

etc.) must be notified, either before or after action is taken. Compliance review may also be required. Deploying an analytics solution effectively requires a well-designed pipeline that takes these considerations into account.

D. Clean architectural separation.

Sooner or later, nearly every organization that deploys analytics solutions discovers that the analytics life cycle does not synchronize well with the product release cycle, i.e. the release cycle for the other system elements that make up the total deployed solution. We will discuss some factors that influence the analytics life cycle momentarily, but from an architectural perspective, a clean separation, or even isolation, is required to support a "plug and play" relationship between the analytics engine and the other system elements.

In most cases, this requires clearly defined interfaces for the analytics engine's inputs and outputs. Input data must be converted to a standard format, so the analytics engine can operate on known data elements of known type. Specific output values must be defined that can be interpreted and acted on by those system elements executing the decision pipeline.

In the ideal world, these interfaces would be fixed and immutable. In practice, only rarely can this be achieved. More practical solutions typically involve configurable interface elements that collect the inputs before analytical processing (sometimes called "gatherers") and convert model outputs to formats expected by the decision pipeline (aka "scatterers"). The entire configuration looks like this:

Raw data → system → gatherers → analytics engine → scatterers → decision pipeline → execution

When this is done well, the gatherers and scatterers do not need to be reworked with every analytics release, but only when key data elements or expected conclusions change.

E. Analytics Life Cycle

Selecting initial model(s) for experimentation, which we have discussed in this paper, is just the first step in an iterative life cycle that will eventually lead to deployment of (hopefully) consistently improving versions of the analytics engine. Here we sketch out the some other steps in the life cycle, which have been dealt with elsewhere (Witten & Frank).

F. Select metrics and thresholds.

In order to evaluate a model, it is necessary to have criteria. Specific metrics must be selected, both positive (detection rate) and negative (e.g. false positives, false negatives). Appropriate thresholds for each metric must be selected. In many cases, both the metrics and the acceptable thresholds are defined by industry standard practice.

G. Evaluate the model(s).

Tests must be run to evaluate model performance and choose the model(s) to be deployed. All models should be tested on the same data, which was not in any way part of the prior selection and/or training process.

H. Deploy model(s).

Selected model(s) are put into production.

I. Monitor and calibrate.

Once a model is put into production, the key metrics should be re-evaluated on actual data. Thresholds that warrant re-training of the model should be defined, and the model re-calibrated when necessary.

J. New models.

The selection of a particular technology or modeling approach is not static. New ideas and new model opportunities are always on the horizon. The advantage of already having a model in production is that subsequent models can be brought up alongside the current production model until the new model is ready for prime time. This is often called the "champion / challenger" approach. Cutover to the challenger model occurs when it consistently outperforms the current champion. Then the process can begin again with a new challenger.

IX. Conclusions and Further Work

In this paper we have outlined a process that fledgling analytics teams can use for selecting one or more models for initial exploration and refinement. We have offered some preliminary evidence which suggests that such an approach may yield reasonable results.

Finally, we have discussed how the initial model should be embedded within an effective analytics architecture, and how it forms the first step of an iterative analytics life cycle that supports continuous improvement.

Future work is required to more rigorously evaluate the decision framework we propose, including both the attributes (questions) and values (answers) assigned to each technique. The approach can also be expanded to include other techniques, especially those that are appropriate for problems other than classification and prediction. Additionally, select the right parameters of execution is another critical point that might an area for extending our work for future.

References

[1] A. Abdallah, M.A. Maarof and A. Zainal, "Fraud detection system: A survey. Journal of Network and Computer Applications," Journal of Network and Computer Applications 68 (2016) 90–113, 2016.

[2] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science, 60* (2015), 708 – 713.

[3] A.F. Atiya, "Bankruptcy prediction for credit risk using neural networks: A survey and new results," *IEEE Transactions on Neural Networks, 12*(4), 929 – 935, 2001.

[4] A.C., Bahnsen, D. Aouada, A. Stojanovic and B. Ottersten, "Feature engineering strategies for credit card fraud detection," Expert Systems With Applications 51 (2016) 134–142.

[5] N. Carneiro, G. Figueira and M. Costa, "A data mining based system for credit-card fraud detection in e-tail," Decision Support Systems 95 (2017) 91–101.

[6] K. Chitra and B. Subashini, "Data mining techniques and its applications in banking sector," *International Journal of Emerging Technology and Advanced Engineering, 3*(8), 219-226, 2013.

[7] T. Chowdhury, K. Ramineni, N. Rahman, M. Krishnan, S. Sharma and J. Wong, "Using Big Data to Understand the Impact of Email on Business," Intel IT White Paper. October 2015, pp. 1-10. [www.intel.com/IT].

[8] A. Gepp, K. Kumar and S. Bhattacharya, "Business failure prediction using decision trees," *Journal of Forecasting, 29*(6), 536–555, 2010.

[9] A.K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognition Letters, 31*(8), 651–666, 2010.

[10] E.Y. Li, "Artificial neural networks and their business applications," *Information & Management, **27*** (1994), 303-313.

[11] R. Li, "Top 10 data mining algorithms, explained. KDnuggets News," 2015. Retrieved on 01/25/2018 from: http://www.kdnuggets.com/2015/05/top-10-data-mining-algorithms-explained.html.

[12] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, "Data mining techniques and applications – A decade review from 2000 to 2011," *Expert Systems with Applications, 39* (2012) 11303–11311.

[13] A. Likas, N. Vlassis and J.J. Verbeek, "The global k-means clustering algorithm," *Pattern Recognition, 36*(2000), 451-461.

[14] H. Lu, R. Setiono and Liu, "Effective data mining using neural networks," *IEEE Transactions on Knowledge and Data Engineering, 8*(6) 957-961, 1996.

[15] J. MacLennan, Z. Tang and B. Crivat, "Data mining with Microsoft® SQL Server® 2008," Wiley Publishing, Inc., Indianapolis, IN 46256, USA. 2009.

[16] A.W. Moore, "Decision Trees," 2012. Retrieved on 01/25/2018 from: http://genome.tugraz.at/MedicalInformatics2/dtree.pdf

[17] A.W. Moore and D. Zuev, "Internet traffic classification using Bayesian analysis techniques," In Proceedings of the 2005 ACM SIGMETRICS International Conference on Measurement and Modeling Computer Systems, Banff, Alberta, Canada, June 6-10, 2005.

[18] M.D. Odom and R. Sharda, "A neural network model for bankruptcy prediction," In *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2, 163-168, 1990.

[19] F.N. Ogwueleka, "Data mining application in credit card fraud detection system," *Journal of Engineering Science and Technology, 6*(3), 311-322, 2011.

[20] D.L. Olson, D. Delen and Y. Meng, "Comparative analysis of data mining methods for bankruptcy prediction," *Decision Support System, 52*(2012), 464-473.

[21] C. Phua, V. Lee, K. Smith and R. Gayler, "A Comprehensive Survey of Data Mining-based Fraud Detection Research," eprint arXiv:1009.6119, Sept. 2010, 1-14.

[22] T.N. Phyu, "Survey of classification techniques in data mining. In Proceedings of the International Multi-Conference of Engineers and Computer Scientists," IMECS 2009, March 18 - 20, 2009, Hong Kong.

[23] N. Rahman, F. Aldhaban and S. Akhter, "Emerging Technologies in Business Intelligence," In Proceedings of the IEEE Portland International Center for Management of Engineering and Technology (PICMET 2013) Conference, San Jose, California, USA, July 28 - August 1, 2013, pp. 542-547.

[24] N. Rahman and S. Iverson, "Big data business intelligence in bank risk analysis," *International Journal of Business Intelligence Research (IJBIR), 6*(2), 55-77, 2015.

[25] P.A. Schlesinger and N. Rahman, "Self-service business intelligence resulting in disruptive technology," *Journal of Computer Information Systems, (JCIS), 56*(1), 11-21, 2015.

[26] Shaw, M.J., Subramaniam, C., Tan, G.W., & Welge, M.E. (2001). Knowledge management and data mining for marketing. *Decision Support Systems, **31*** 2001 127–137.

[27] K.-S. Shin, T.S. Lee and H.-J. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications, 28*(1), 127–135, 2005.

[28] L. Shu-Hsien, C. Pei-Hui and H. Pei-Yuan, "Data mining techniques and applications- A decade review from 2000 to 2011," *Journal of expert system with applications, 39*(2012), 11303–11311. http://dx.doi.org/10.1016/j.eswa.2012.02.063

[29] S. Tong and D. Koller, "Support vector machine active learning with applications to text classification," Journal of Machine Learning Research, (2001) 45-66.

[30] J. West and M. Bhattacharya, "Intelligent financial fraud detection: A comprehensive review. Computer & Security 57, 47-66, 2016.

[31] R.L. Wilson and R. Sharda, "Bankruptcy prediction using neural networks," *Decision Support Systems, 11*(5), 545-557, 1994.

[32] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z. Zhou, M. Steinbach, D.J. Hand and D. Steinberg, "Top 10 algorithms in data mining, *Knowl Inf Syst* (2008), 14, 1-37.

[33] D.W. Hosmer Jr., S. Lemeshow and R.X. Sturdivant, "Applied logistic regression," 3rd Edition, Wiley, USA. 2013.

[34] E. A. Lopez-Rojas and S. Axelsson, "BankSim: A Bank Payments Simulator for Fraud Detection Research," *The 26th European Modeling and Simulation Symposium*, Sep. 2014.

[35] "Adaptive real-time machine learning for credit card fraud detection," Adaptive real-time machine learning for credit card fraud detection | Machine Learning Group. [Online]. Available: http://mlg.ulb.ac.be/ARTML. [Accessed: 24-Jan-2018].

[36] N. Rahman, A Taxonomy of Data Mining Problems. *International Journal of Business Analytics (IJBAN), 5*(2), 73-86, 2018.

APENDIX A – THE DECISION MATRIX

	SUP	UNSUP	#OBS	#VAR	TYPV	QUALV
NN	Most	Few	Large	Large	Numeric only	Very Tolerant
KM	None	All	Very Large	Moderate	Numeric only	Slightly Tolerant
REG	Most	Few	Very Large	Few	Numeric only	Slightly Tolerant
CART	All	None	Large	Large	Both	Very Tolerant
DT	Most	Few	Large	Very Large	Both	Very Tolerant
KNN	All	None	Moderate	Moderate	Numeric only	Slightly Tolerant
SVM	All	None	Moderate	Very Large	Numeric only	Moderately Tolerant
SCA	All	None	Moderate	Moderate	Both	Intolerant
RF	All	None	Large	Very Large	Both	Very Tolerant

	MISSV	STAND	#DV	QDV	SDV	PATTERN
NN	Intolerant	No Impact	One	Very High	Speed Impact	No Impact
KM	Moderately Tolerant	No Impact	Many	High	Moderate Impact	Moderate Impact
REG	Moderately Tolerant	Significant Improvement	One	Low	??	Depends on regression form
CART	Moderately Tolerant	No Impact	Many	??	No Impact	No Impact
DT	Very Tolerant	Significant Improvement	Many	??	??	No Impact
KNN	Moderately Tolerant	No Impact	Many	Moderate	Significant Impact	Moderate Impact
SVM	Intolerant	No Impact	One	Low	Requirement	Slight Impact
SCA	Very Tolerant	No Impact	One	Moderate	Speed Impact	Strong Impact
RF	Very Tolerant	No Impact	Many	??	Moderate Impact	Strong Impact

	OVERF	2NDORD	EXPL	VIS
NN	No control	None	No support	N
KM	Number of clusters	K means, K variances	No support	Dimensional cross-sections
REG	Number of independent variables	R squared, T statistics	Variable weights (assuming normalized values)	Y
CART	Number of split nodes, Depth of trees.	None	Graphical support	Y
DT	Number of split nodes, Depth of trees.	Branching statistics, Tree depth	Graphical support	Y
KNN	Parameter K	K means, K variances	Graphical support	Y
SVM	Outlier rejection.	Margin	Graphical support	Dimensional cross-sections
SCA	Restrict contribution patterns	Lift statistics	Strong support	N
RF	Number of trees, Branching limits, Tree depth	Number of trees, Branching statistics, Tree depth	Graphical support	Y

APENDIX B – R PACKAGES AND FUNCTIONS

Name	Explanation
library(mice)	Used to show the missing data
library(VIM)	Used to show the missing data
library(caTools)	Used to split the data into training and testing sets
library(neuralnet)	Used for analyzing Neural Networks technique
library(MASS)	Used for analyzing Regression technique
library(randomForest)	Used for analyzing Random Forest technique
library(e1071)	Used for analyzing Support Vector Machines technique
library(class)	Used for analyzing K-Nearest Neighbors technique
library(gmodels)	Used for analyzing K-Nearest Neighbors technique
library(cluster)	Used for analyzing K-Means technique
library(rpart)	Used for analyzing Classification and Regression Tree technique
library(rpart.plot)	Used for analyzing Classification and Regression Tree technique
library(caret)	Used for analyzing Classification and Regression Tree technique
library(rpart)	Used for analyzing Decision Trees technique
read.csv	Read the file of the dataset and upload it to R Environment
head	Shows the first X numbers of rows of the dataset
predict	For predictions from the results of various models
svm	Used to train a support vector machine
sum	Returns the sum of all the values present
rpart	Used to train a Classification and Regression Tree
kmeans	Perform k-means clustering on a data matrix
crossTable	An implementation of a cross-tabulation function
knn	k-nearest neighbor classification for test set from training set
randomForest	Implements Breiman's random forest algorithm
glm	Used to fit generalized linear models
step	Select a formula-based model by AIC
compute	Stores results in a remote temporary table
neuralnet	Used to train neural networks using backpropagation
apply	Returns a vector or array or list of values obtained
sample.split	Split data from vector Y into two sets in predefined ratio
subset	Return subsets of vectors
aggr	Calculate or plot the amount of missing/imputed values
md.pattern	Display missing-data patterns