

Is This Brand Ephemeral? A Multivariate Tree-based Decision Analysis of New Product Sustainability

INTRODUCTION

Globalization, technological innovation, advances in supply chain management, and perhaps most importantly the diversification of consumer needs have led to the proliferation of new brands of consumer goods and services. Every year in the U.S. market approximately 1,000 new types of ice cream and other dairy products are launched (New Product News, 1999). In the consumer service industry, one of the leading credit card issuers, Capital One, issues over 1,000 new types of credit cards every year to target specific segments of the consumer market (Fishman, 1999). The proliferation of new consumer goods and services is not only restricted to developed countries. In China, for example, for categories such as packaged food (e.g., bottled water) and popular electronics equipment (e.g., mobile phones and Discmans) a substantial number of new brands appear in the market every month—while an equally large number of brands vanish every month. Because of the relatively low overhead for manufacturers to produce goods that are slightly differentiated from existing products, it is common practice for manufacturers to launch multiple items under one brand name, multiple lines of brands, or multiple lines of products under the same or different brand names. This is often used by companies as a defensive strategy in protecting existing market, or as a proactive approach in opening untapped market segments. While some of the

new launches will successfully meet the preference and taste of specific market segments and survive the initial launching phase, the reality is that others will fail and prove to be ephemeral. Indeed, for consumer packaged goods such as snacks and stationery, it is estimated that only one-quarter to one-third of new launches will survive more than two years (Kotler, 2000).

The commoditization of consumer goods and the proliferation of new brands have increasingly shifted the focus of the decision concerning new product launches from the supply chain to the so-called “demand chain” (Kahn, McLister, 1997). In order to quickly adjust to changing consumer tastes and preferences, information about consumer reception of a new launch has to be gathered quickly, and a timely assessment of the product must be made before further resources are committed. Because shelf space in retail outlets is limited, new launches generally do not enjoy an extended period of time to prove themselves. Managers are pressured to make a quick decision as to whether a product should be allowed to remain in the market. In many cases, these decisions are based on limited performance data from the initial period of product launching. Prolonged decisions to discontinue unsuccessful products may unnecessarily lead to increased losses and to the wasting of resources that otherwise could have been used for launching new brands or promoting more promising existing brands. On the other hand, prematurely terminating potentially successful products would mean lost opportunities and profits.

In this paper, we propose a method that aims to help managers determine whether a newly launched product should be allowed to remain in the market given only early performance data from the initial phase of the launch. The proposed method is based on a data-driven, multivariate decision-tree approach, and is tailored to the action requirements of decision science applications. Organizations have now extensively used data-driven, learning-based systems to support their decision-making process, and many have reported significant financial gains as a result of such improvement (e.g., Goonatilake, 1995). To illustrate the proposed method, we apply it to transaction data from a popular consumer packaged good—the instant noodle cup—which are gathered from its largest market, which is the Japanese market. Conventional wisdom suggests that the decision to continue or terminate a new product can be based on aggregated sales volume or profit data gathered during the launching period. However, our experience with the packaged food industry in Japan was that such a method might fail. For example, our analysis indicated that in the instant noodle cup market a new launch typically can only achieve a small market share. When a new product has high trial volume (the first purchase) but low repeat volume (the second and future purchases), the product is more likely to fail than one that has smaller trial volume but high repeat rates. A decision that solely relies on early sales volume may identify a potentially weak product that has

good initial volume (perhaps because of promotional cuts) as a success and a potentially strong product that does not have high initial volume as a failure.

As far as we know, the literature on decision analysis for continuing or terminating new products in a mature market is sparse. The most recent example that we found is Lin & Chen (2004). The authors described a fuzzy-logic based decision system that was used to support a go/no go business decision for launching a new product. There are, however, several related literatures—managing new, innovative products and brands; demand forecasting; and new product adoption. The first literature mainly concerns the development and introduction of innovative and often radically designed new products into a market, either traditional or new (Cohen, Eliashberg & Ho, 1996; Crawford & Benedetto, 2003). Thus, their focus is not on the point of decision after the initial launch of undifferentiated products in a mature market. On the other hand, demand forecast models tend to study demand under the framework of predictive models, and models for repeat purchase (Fourt et al., 1960; for Japanese market application, Nakanishi, 1984). Some other recent examples include Cooper et. al. (1999) where a knowledge discovery system for tactical planning forecast was proposed. Although companies do require accurate forecast models to predict sales during and after the launching phase, this specific information is not particularly helpful for predicting the fate of a new launch. Finally, new product adoption literature aims at predicting the trajectory of sales of new products and

their diffusion after they are introduced in the market (Bass, 1969; Kalish, 1985; Mahajan, Muller, & Wind, 2000). All three bodies of literature, while related to our current interest, are not immediately relevant to solving the problems we encountered.

There are several distinctive features about our approach. First, instead of requiring a forecast model for predicting sales volume for future periods, the approach provides an action-oriented decision tool to help managers arrive at and, perhaps to a lesser extent, justify a “go or not go” decision. Second, it integrates information from several sources—product, customer, and market conditions. Furthermore, the tool allows distinct types of input. For example, it can simultaneously handle sequential pattern data such as sales volume as a function of time and other non-time-varying input. Accordingly, it generates a set of interpretable multivariate induction rules as output.

MANAGING NEW LAUNCHES

In this section, we describe the specific example under which the decision problem arises—the instant noodle cup market in Japan, which also happens to be the largest market for instant cup noodles. Although the example contains conditions that are rather specific to the instant noodle industry, the problem we illustrate should be rather general in nature. Providers of other consumer products (e.g., snacks) and services (e.g., credit cards) face similar problems,

although in different contexts (e.g., the launching and observation period may be longer).

The Japanese instant noodle market is the largest in the world, and is approximately U.S. \$4 billion in size. Six major companies—among them Nissin Food, which first launched packaged instant noodles—controls approximately 90% of the market (Yano, 2002). Each manufacturer maintains several lines of products under different brand names. The instant noodles are usually differentiated by factors such as price (a unit sells for U.S. 70 cents to \$2.00 in Japan), flavor (traditional chicken and beef tastes to newer ones such as green tea and cheese-curry), type (e.g., ramen, udon), and package (e.g., cup, bowl, pillow-like package). In this paper, we focus on instant cup noodles. While the number of new launches differs from year to year, our recent survey showed that there were over 300 new products of instant cup noodles introduced in the year 2001. There are several primary retail channels for instant cup noodles in Japan, including supermarkets, convenience stores, and local grocery stores. Shelf space differs across channels and generally is highly competitive. For example, the shelf space available in a supermarket for instant noodles allows the display of 40 to 50 products, while in convenience stores such as Seven-Eleven™, the competition is more intense, and only 20 to 30 products may be displayed. Because of the competition for shelf space, brand managers often have to constantly review performance data and make decisions about which products

are to be displayed in each market. More importantly, they are also required to make judicious decisions as to whether or not a newly launched product should be allowed to continue.

The traditional method for making this decision is often based on a combination of aggregated sales data and the experience of brand managers. For example, when early aggregated sales volume is not satisfactory, the brand will be terminated. However, this approach is dependent upon the experience of the individual manager. The decision is also complicated by the fact that for most newly launched products the initial sales volumes are generally small and contain high variance. A seemingly unpopular product in the first two weeks may suddenly gather momentum because it takes time for word of mouth to spread. On the other hand, a product may be seemingly popular in the first two weeks because of the initial promotional price cut—a factor that often confounds the performance of the product during the initial launching period—but fail to sustain its popularity when the price promotion is stopped. Furthermore, aggregated sales data do not take into account the characteristics of who is buying (or not buying) the product. For example, when a large proportion of the consumers are trial buyers and only buy because of the initial price promotion, using only aggregated data on sales volume could be misleading.

It is also important to incorporate marketing variables into the decision. Some new items are merely extensions of an existing, strong brand line, and

some are designed as flanker brands targeting a different group of consumers from the established brand. In such cases, the strength of the existing brand or the reputation of a manufacturer may play a role in increasing the likelihood of survival. Some other marketing variables are more complex. For example, in order to develop new flavors for instant cup noodles, manufacturers often scout famous local restaurants for recipes. Once they have found a marketable recipe, the manufacturer licenses the brand name of the selected restaurant and uses it as the brand name for their new product. Some marketers also use the names of local places as brand names (this is common practice in the bottled water industry). Besides a few brand names that have broad reach, most instant cup noodle brands appeal to local tastes and preferences and will likely survive if the products can sell well in a concentrated region. Such marketing variables, which could enhance the quality of decisions, are not made explicit in traditional analysis.

The problem we describe above presents a challenge and an opportunity for decision science researchers and practitioners. To satisfactorily solve the problem, efficient data learning systems that incorporate comprehensive information need to be developed. In this paper, we describe a tree-based decision system that directly addresses the shortcomings of traditional methods, and we compare its performance to several commonly used algorithms.

THE DECISION SYSTEM

This section describes the decision support system and the data set. Comprising three components, the decision system analyzes historical data and provides interpretable results for supporting marketing decisions. The first component is a preprocessing machine, which extracts relevant input variables from raw transaction data collected from frequent shopper programs (FSP) from various retail channels. The input variables are then relayed to the second component, the knowledge discovery machine EBONSAI, which contains two subcomponents: the Data Transformation Engine (DTE) and the Inference Engine (IE). The DTE transforms temporal data into string sequence patterns, and the IE attempts to learn from the sequence and determine a set of induction rules. Finally, the reporting component reports the set of induction rules. Figure 1 depicts the decision system.

Insert Figure 1 Here

Data Description

The raw transaction data sample collected from our FSP system contains 38 million transactions from 43,363 customers. The data set includes purchase history collected by a system of seven retail stores in a supermarket store chain from August 2000 to October 2001. At least 3 weeks of data were collected after the release of each product, and there were a total of 579 new products of instant

noodle introduced during this period of time. Figure 2 shows the distribution of the survival time for this sample of new launches. It can be seen that the new launches approximately cluster into several classes :

- Class I. These products quickly disappeared from the shelf within 30 days after launching.
- Class II. These products survived for 2 to 4 months and eventually vanished from the shelves.
- Class III. These products lingered on for 5 to 8 months.
- Class IV. These products survived for more than eight months. The products in class are regarded as successful launches.

Insert Figure 2 here.

A portion of products from Class I in fact consisted of test-market cup noodles. Test market products were only launched to gather pilot data on how the market would respond to an experimental new product. By design, test market products were not intended to last, and so this class of products was excluded in the subsequent analysis. In order to focus our analysis on surviving and non-surviving brands, we collapsed Class II and III and defined the combined class as non-survivors, whereas Class IV was defined as survivors. The variable

that indicated survival status was denoted by SURV. This variable will be used as our primary outcome variable in the subsequent analysis.

We further partitioned the sample to form the training and the test data sets, respectively. Table 1 shows the number of products cross-tabulated by the class variable and by whether it belongs to the training or the test set. The decision system was first trained on the training data set to produce an EBONSAI tree-based model. Then the performance of the model was evaluated on the hold-out sample of test data set. This blinded approach to evaluation ensures an objective assessment of how the system may perform with a new and unseen data set.

Target	Test	Train	Total
Non-survivors	84	163	247
Survivors	25	175	200
Total	109	338	447

Table 1. Distribution of surviving products in training and test sets

To predict survival, three sources of information—customer information, product attributes, and marketing information—were used. Customer information was extracted from the FSP raw data by the preprocessing machine. It included the variables REPR and RHUSER, which respectively represented the repeat purchase rate and the ratio of heavy instant cup noodle users. REPR was defined as the proportion of buyers who had purchased the product in the previous week and also bought in the current week. A heavy user was defined as someone who fell into the top 33% of the buyers by volume (number of units). Both variables are time-varying (i.e., these variables changed over time). Specifically, they were coded as a time sequence of alphabets, with each alphabet indicating the status at a specific time point. The second source of information, product attributes, included the variables MANUF and TASTE. They were categorical variables respectively representing the manufacturer of the product and the flavor of the product, and both were non-time-varying. The third category of variables was related to sales and pricing: SALES was weekly sales volume of each product, which was a time-varying variable; REDPRICE was the largest discount ratio of sales price to regular price for the first 3 weeks after the release date (value: between 0 to 1); and DISC was the time at which the product was first sold with discounted price (value: 1st week, 2nd week or 3rd week). Table 2 shows the list of important input variables to the decision system and their summary statistics.

Except for the variables REPR and RHUSER, the correlations among the other variables were not significant. The correlation between REPR and RHUSER was approximately 0.48 (both measured at the 3rd week).

Variables	Average	Standard Deviation	Data Type
SALES(1)	3.01	1.41	Time-varying
SALES(2)	3.07	1.40	Time-varying
SALES(3)	3.05	1.41	Time-varying
REDPRICE	0.94	0.12	Non-time-varying 0-1
DISC	0.56	1.03	Non-time-varying (1,2,3)
REPR(2)	2.10	1.06	Time-varying
REPR(3)	1.86	1.16	Time-varying
RHUSER(1)	2.99	1.42	Time-varying
RHUSER(2)	2.66	1.62	Time-varying
RHUSER(3)	2.46	1.74	Time-varying

Table 2. Descriptive statistics of important variables in the model. The week number is represented as (1), (2), and (3).

Knowledge Discovery Engine : EBONSAI

EBONSAI, or Extended-BONSAI is a tree-based rule-induction engine that classifies pattern data in the form of character substrings. EBONSAI extends the work of BONSAI, which was first developed by Shimozono et al. (1994). The underlying concept of BONSAI originates from Arikawa et al., 1993, in which the authors applied the idea to the identification of genome sequences. Effectively, EBONSAI is a multi-attribute version of a commonly used induction method, ID3, of which later versions are known by other names such as C4.5 and C5.0 (Quilian, 1986, 1993). A rule-induction engine achieves automatic learning from a set of examples (called a training or learning set) for which the outcome is known. With sufficient examples, the system generates induction rules that mimic the decision outcome of the examples in the training set. Many tree-based rule-induction methods, including ID3 and EBONSAI, create a tree from a series of binary splits on attributes. The collection of splits partitions the attribute space into a set of non-overlapping rectangles.

The unique feature of EBONSAI is its treatment of string variables (e.g., weekly observations of an outcome such as a high or low pattern of sales). Instead of treating a string as comprising individual variables in splitting, EBONSAI seeks “character substring” to optimize splits (Hamuro, et al., 2002). An example of a character substring is an observed sales pattern for a five-week period: (VH,H,M,L,L), in which we first see a very high level of sales (VH) in the first

week, then a high-level of sales (H) in the second week, then moderate sales (M) in the third week, and finally low sales (L) in the fourth and fifth weeks. Now, we see a high proportion of failure cases containing the pattern (VH,H,M,L,L), then a learning algorithm may want to include the string as a criterion for splitting the data—namely, partitioning the data into those that contain the pattern and those that do not. In general, EBONSAI treats patterns as strings of categories, and the collection of categories is called the alphabet set. For example, {VH,H,M,L} forms an alphabet set. Details of the learning algorithm will be provided in the following subsections.

EBONSAI comprises two engines—the data transformation engine (DTE) and the knowledge discovery engine (KDE). DTE transforms data into the appropriate character strings suitable for input into KDE. Then KDE applies a greedy algorithm to search over the space that contains combinations of character strings to identify the appropriate candidate for splitting the data space. We separately describe the two engines as follows:

Data Transformation Engine

DTE transforms pattern data into character strings. The term “pattern data” refers to time-varying input: SALES, REPR, and RHUSER. The variable SALES, for example, is a d-tuple (from d weeks of sales data) and is required to be transformed into a string of characters in which each character represents a

category of level of sales. The transformation takes place in two steps: first, each continuous variate is discretized into k levels, where k is predetermined. Second, the d -tuple of the continuous variates is mapped to a corresponding d -tuple of the categorical variable. Note that the order of the d -tuple is important and that each d -tuple will be treated as multivariate data in the implementation of the inference engine. Figure 3 illustrates the two-step transformation procedure of DTS.

Insert Figure 3 Here

We use the following method to determine the number of categories k in discretizing a continuous variable. First, we use $k = 3, 5$ for each time-varying input. As a result, there are altogether 2^d possible ways to create a transformed data set, which we called the k -subsample. Second, we apply the learning algorithm to each of the 2^d k -subsamples and select the best combination of k according to the criterion of predictive accuracy in the test set. As we shall see later, the result from EBONSAI is rather robust with regard to the choice of the value of k .

Inference Engine

KDE is designed to learn from the data (the training set) a set of induction rules K so that K can be used to classify new observations. EBONSAI is distinguished from other tree-based methods in its ability to process pattern strings. When the attributes are not strings, EBONSAI is functionally similar to ID3. However, when some of the variables contain strings such as five-week sales volume (VH,H,M,L,L), EBONSAI performs differently from ID3 in that it will search for candidate substrings for an optimal split. Generally, the space of candidate substrings is quite large (e.g., besides (VH,H,M,L,L) there may be other candidate substrings such as (H, VH, L, M, M) that are eligible to compete for being the optimal split. EBONSAI applies a greedy algorithm to select the optimal split among a pool of possible combinations of substring patterns, and the algorithm then recursively searches for optimal splits in the subsequently partitioned data spaces. As with other tree-based algorithms, one can either terminate splitting when a predetermined criterion is reached or prune an overly grown tree by first allowing the splitting process to continue until the recursively partitioned data spaces become very small. However, unlike traditional classifying learning algorithms such as ID3 and CART (Breiman et al., 1984), which recursively partition the data space by univariate splits, EBONSAI splits the data space by combinations of substring patterns of varying lengths.

Specifically, let P and N , respectively, denote the set of positive and negative examples. In the current application, P is the set of surviving brands, and N is the set of failed brands. A regular pattern π is defined as a string $\alpha_0 x_1 \alpha_1 x_2 \cdots x_k \alpha_k$, where each α_i is a constant substring and each x_i is a variable that matches any string. Thus, the pattern $\alpha_0 x_1 \alpha_1 x_2 \cdots x_k \alpha_k$ represents a string that contains the specific substrings $\alpha_0, \alpha_1, \dots, \alpha_k$, and in that order. Each α_i is some substring in the set $P \cup N$. For any regular pattern π generated from P and N the cost function $E(\pi)$ is given by

$$E(\pi) = \frac{p_1 + n_1}{|P| + |N|} I(p_1, n_1) + \frac{p_0 + n_0}{|P| + |N|} I(p_0, n_0), \quad (1)$$

where $p_1(n_1)$ denotes the number of positive examples in $P(N)$ that match π , $p_0(n_0)$ denotes the number of positive examples in $P(N)$ that do not match π , and

$$I(p, n) = \frac{p}{p+n} \log \frac{p}{p+n} - \frac{n}{p+n} \log \frac{n}{p+n}, \quad (2)$$

and 0 if either p or n is 0. The cost function is similar to the Gini or the entropy function, which are used in CART and other tree-based methods. They basically measure how well a split performs in separating the examples into homogeneous groups. EBONSAI uses a greedy algorithm to search for a pattern π that minimizes the cost $E(\pi)$ at a node.

Because the set of all possible strings can get extremely large, EBONSAI employs an alphabet indexing procedure to simplify the search. Alphabet

indexing is a mapping from an alphabet that contains a large number of symbols into another alphabet with fewer symbols without losing any information for classifying positive and negative examples. The indexing procedure employs a leap-and-bound search algorithm that starts by randomly selecting two small subsets of positive and negative examples and an index that is randomly generated—e.g., (VH, H, M, L,L) can be given an index (1,0,0,0,0). Then the algorithm searches a new index (by randomly flipping the 1’s and 0’s) from its neighborhood in such a way that the new index has a score that is best within that neighborhood (Hamuro et al., 2003). The score is defined in a way to reflect the average success rates of classifying the positive and negative examples.

With the flexibility in directly handling strings, EBONSAI can incorporate substrings of the form $^{\alpha}$ and $\alpha\$,$ which respectively represent the initiation and the termination of the sequence α . This feature, which is not present in the original BONSAI, can be quite powerful. For example, a certain pattern appearing at the beginning of the launching period may have greater predictive power than the same pattern appearing at other positions.

Statistically speaking, EBONSAI uses multivariate split to preserve positional information and possible interaction between variables. This has an important advantage over systems that are based upon univariate split, which is implemented in machine learning algorithms such as C5.0 and CART. We provide a simple example to illustrate how character string-based systems can overcome

a shortcoming in univariate systems. A similar example is used in (Giuffrida, Chu, & Hanssens, 2000). Table 3 shows a simple data set that contains the variable on sales volume of the first two weeks—SALES1 and SALES2—and the target variable SURV (0 = not survive, 1 = survive). An inspection of the data reveals that neither SALES1 nor SALES2 alone has strong predictive power. However, there is strong interaction between the two variables. A strong rule exists within the data: “If SALES1 = L and SALES2 = L, then SURV = 0.” When EBONSAI was applied to the data set, it successfully recovered the strong rule. Figure 4(a) shows the tree from EBONSAI induced by the data. Algorithms that are based on univariate split cannot exactly recover the strong rule—they may determine that no split can improve the performance of the tree or rely purely on univariate splits to recover the rule. Figure 4(b) shows the tree obtained via ID3 (Giuffrida, Chu, & Hanssens, 2000). Indeed, when the dimensionality of the multivariate problem increases, the interaction pattern between variables will become increasingly complex. Accordingly, the univariate approach will become even less powerful in detecting and summarizing potentially strong or interesting rules. EBONSAI is designed to overcome this difficulty.

SALES1	SALES2	SURV
L	H	1
L	M	1
L	VH	1
VH	L	1
H	L	1
M	L	1
L	L	0
L	L	0
L	L	0

Table 3. An example data set to illustrate the shortcoming of learning based on univariate split

INSERT FIGURE 4(a) and (b) HERE

RESULTS AND COMPARISON WITH OTHER METHODS

The decision tree that EBONSAI produced from the training data set contained 16 nodes. Both the variable repeat purchase rate and the ratio of heavy cup users occurred multiple times in the tree and thus are seen to be critical in predicting

the survival of the brand. For example, if the ratio of heavy users consistently decreases in the three-week period, then the brand is likely to fail.

We evaluated the performance of EBONSAI by several measures of accuracy, and the result is given in Table 4. Predictive accuracy is defined as the proportion of correctly classified brand for $SURV = 1$ (and $SURV = 0$)—i.e., the number of corrected classified statuses (survive or not survive) divided by the total number of products with that status. Overall accuracy is defined as the total number of correctly classified brands divided by the total number of brands in the sample.

	Training Set	Test Set
Predictive accuracy for survivors	97.7	84.0
Predictive accuracy for non-survivors	60.3	32.1
Overall predictive accuracy	75.1	48.0

Table 4. Accuracy of EBONSAI for the training set and the test set

The results partly reflect the choice of the cost matrix. After consulting with the marketing experts in the industry with regard to misclassifying a surviving brand as a non-survivor, we imposed a penalty three times as large as

for misclassifying a non-surviving brand as a survivor. This skewed the decision tree toward tending to include more predicted survivors.

To further examine the robustness of the tree, we randomly split the sample into two halves and compared the trees that each half produced. We found that the tree structures appeared to be rather similar, as were their classification rates

We also compared the predictive accuracy between EBONSAI and three other commonly used machine learning methods: C5.0, CART, and neural network. Comparisons between neural network and decision tree methods have been reported in the decision science and machine learning literature (e.g., Bhattacharyya, 1998; Liu Sheng et al., 2000). We evaluated the accuracies of the four methods on both the training and the test sets. For neural network, we used an architecture with one hidden layer with 20 nodes and the back-propagation algorithm for training. The neural network procedure was implemented in Clementine.

Figure 5 displays the prediction accuracies of EBONSAI and the other methods on both the training set and the test set. Recall that we created different k -sub-samples to evaluate the effect of discretizing the continuous-variable sales volume into k levels ($k = 3, 5$). The label for each set of bar chart represents how the value of k over the time span. The overall prediction accuracy of EBONSAI for the training data set is similar to those of the other three methods (Figure 5a).

However, the overall predictive accuracy of EBONSAI for test data is significantly higher than for any of the other three methods (Figure 5b).

It appears that both C5.0 and CART exhibit some overfitting to the training data set. C5.0, in particular, tends to be sensitive to how the continuous variable has been discretized. For EBONSAI, the results appeared to be rather consistent across the various sub-samples, which implies that the performance of the proposed procedure is rather robust with regard to variation in the way the continuous variable was discretized.

In order to see how the prediction accuracy of EBONSAI varies by changing its pruning parameter, we performed some experiments. Pruning parameters are used to indicate the complexity of a tree model. To find an optimal pruning parameter, one needs to solve the trade-off between pruning the tree too much (resulting in the introduction of bias) and pruning too little (resulting in excessive variance). Using C5.0 as a benchmark for comparison, we varied the pruning parameter (cut rate) of both trees. For EBONSAI, we pruned the tree through assessing the binomial probability of misclassifications within the set of cases presented to the node by varying the minimum allowable number of cases falling into a node. Figures 6 (a) and (b) illustrate the change in the prediction accuracy (for survivors) of EBONSAI and C5.0 for the training and test sets with respect to the change of pruning parameters. Figure 6 suggests that EBONSAI maintains a high level of accuracy with a range of pruning parameters and performs quite

well relative to C5.0 in terms of robustness. We note that accuracy for EBONSAI remains constant for a range of pruning parameters for the training data. This could be an artifact of the possible ceiling effect for predictive accuracy.

CONCLUSION

Companies that provide consumer goods and services are facing tremendous challenges. The marketplace is becoming increasingly commoditized, with fast-changing market conditions as competitors continually enter and exit the market. Managers and marketers have to make decisions under time constraints and often without full information. To manage the risk that is associated with the uncertain outcomes, a disciplined approach to decision making, based on data and modern decision tools, can be more profitable than relying on ad hoc methods or experience alone. In this paper, we showed how detailed data that are routinely collected from business transactions, coupled with innovative technology, can be combined to help managers arrive at decisions concerning the sustainability of newly launched consumer products. An important feature of our approach is the use of string patterns for tree growing, which has demonstrated that it can lead to improvement over methods that are based on univariate splits. Like other tree-based methods, our approach provides managers with a set of

interpretable induction rules. While our analyses are directed toward a specific industry—instant cup noodles—our methodology is rather general and can be easily adapted to other industries of consumer products.

A limitation of the study reported in this paper is that we have neither included new product development process variables nor marketing strategy variables. Both of these may provide insight into why a new product will succeed. For example, how a company originates and manages the new product development process may create a strategic advantage in the product's long-term success (Ettlie & Subramaniam, 2004). Further, competitive forces in the market will also shape the destiny of a new launch (Hultink & Langerak, 2002). In a future study, we plan to design methods to measure these attributes and incorporate them into the decision model.

REFERENCES

- Arikawa, S., Miyano, S., Shinohara, A., Kuhara, S., Mukouchi, Y. & Shinohara, T. (1993) A machine discovery from amino acid sequences by decision trees over regular patterns. *New Generation Computing*, 11, 361–375.
- F. M. Bass, F.M. (1969). A new product growth for consumer durables. *Management Science*, 15, 215-227.

Bhattacharyya, S., & Pendharker, P.C. (1998). Inductive, evolutionary, and neural computing techniques for discrimination: A comparative study. *Decision Science*, 29, 871-899.

Breiman, L., Friedman, J. H., Olshen, R., & Stone, C. (1984). *Classification and regression trees*. Monterey, CA: Wadsworth and Brooks.

Cohen, M. A., Eliashberg, J., & Ho, T. H. (1996). New product development: The performance and time to market trade-off. *Management Science* 42(2): 173–186.

Cooper, L. Baron, P., Levy, W., Swisher, M., & Gogos, P. (1999). PromoCast: A new forecasting method for promotion planning. *Marketing Science*, 18(3) 301–316.

Crawford, M., & Di Benedetto, A. (2003). *New products management* (7th Ed.). New York: Irwin/McGraw Hill.

Ettlie, J.E., Subramaniam, M. (2004). Changing strategies and tactics for new product development. *Journal of Product Innovation Management*, 21, 95-109.

Fishman, C., This is a marketing revolution. *Fast Company*, May 1999, 206–218.

Fourt, L.A., & Woodlock, J.W. (1960). Early prediction of market success for new grocery products, *Journal of Marketing*, 25(2), 30–38.

Giuffrida, G., Chu, W., & Hanssens, D.M. (2000). Mining classification rules from data sets with large number of multi-value attributes. In C. Zaniolo, P.C. Lockmann, M.H. Scholl, & T. Grust (Eds.), *Advances in database technology*. Berlin: Springer, 335–350.

Goonatilake, S. (1995). Intelligent systems for finance and business: An overview. In S. Goonatilake & P. Treleaven (Eds.), *Intelligent systems for finance and business*. New York: John Wiley & Sons.

Hamuro, Y., Kawata, H., Katoh, N., & Yada, K. (2002). A machine learning algorithm for analyzing string patterns helps to discover simple and interpretable business rules from purchase history. *Progress in Discovery Science, LNAI 2281*, 565–575.

Hamuro, Y., Katoh, N., Ip, E.H., Cheung, S., & Yada, K. (2003). Combining information fusion with string pattern analysis: A new method for predicting

future purchase behavior. In C. Torra (Ed.), *Information fusion in data mining*. Berlin: Springer-Verlag.

Hultink, E.J., & Langerak, F. (2002). Launch decision and competitive reactions: An exploratory market signaling study. *Journal of Product Innovation Management*, 19, 199-212.

Kahn, E.B., & McLister, L. (1997). *Grocery revolution: The new focus on the consumer*. New York: Addison Wesley.

Kalish, S. (1985). A new product adoption model with pricing, advertising and uncertainty. *Management Science*, 31, 1569–1585.

Kotler, P. (2000). *Marketing management*. New York: Prentice Hall.

Lin, C.T., Cheng, C.J. (2004). A fuzzy-logic based approach for new product go/no go decision at the front-end. *IEEE Transaction on Systems, Man & Cybernetics*, Part A, 34(1), 132-142.

Liu Sheng, O.R., Wei, C.P. , Hu, P.J., Chang, N. (2000). Automated learning of patient image retrieval knowledge: Neural networks versus inductive decision trees. *Decision Support Systems*, 20, 105–124.

Mahajan, V., Muller, E., & Wind, Y.. (2000). *New product diffusion model*. Boston & Dordrecht: Kluwer Academic Press

Nakamura, H. (2001). *Marketing of new products..* Japan: Chuokeizai-sha.

Nakanishi, M. (1973) Advertising and promotion effects on consumer response to new products. *Journal of Marketing Research*, 10, 242–249.

Nakanishi, M. (1984). *Frontier in consumer behavior analysis*. Japan: Seibundo-Shinkosha.

New Product News, (1999). Health Benefits Included.

<http://www.preparedfoods.com/archives/1999/9904/9904northamer.htm#efforts>.

Quinlan, J.R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.

Quinlan, J.R. (1993). *C4.5: Programs for machine learning*. (City): Morgan Kaufman.

Shimozono, S., Shinohara, A., Shinohara, T., Miyano, S., Kuhara, S., & Arikawa, S. (1994). *Knowledge acquisition from amino acid sequences by machine learning system bonsai*. Trans. Information Processing Society of Japan. Vol. 35, 2009–2018.

Yano Research Institute, International Business Development Department (2002). *Market share in Japan*. Tokyo, Japan: Yano Research Institute Ltd.

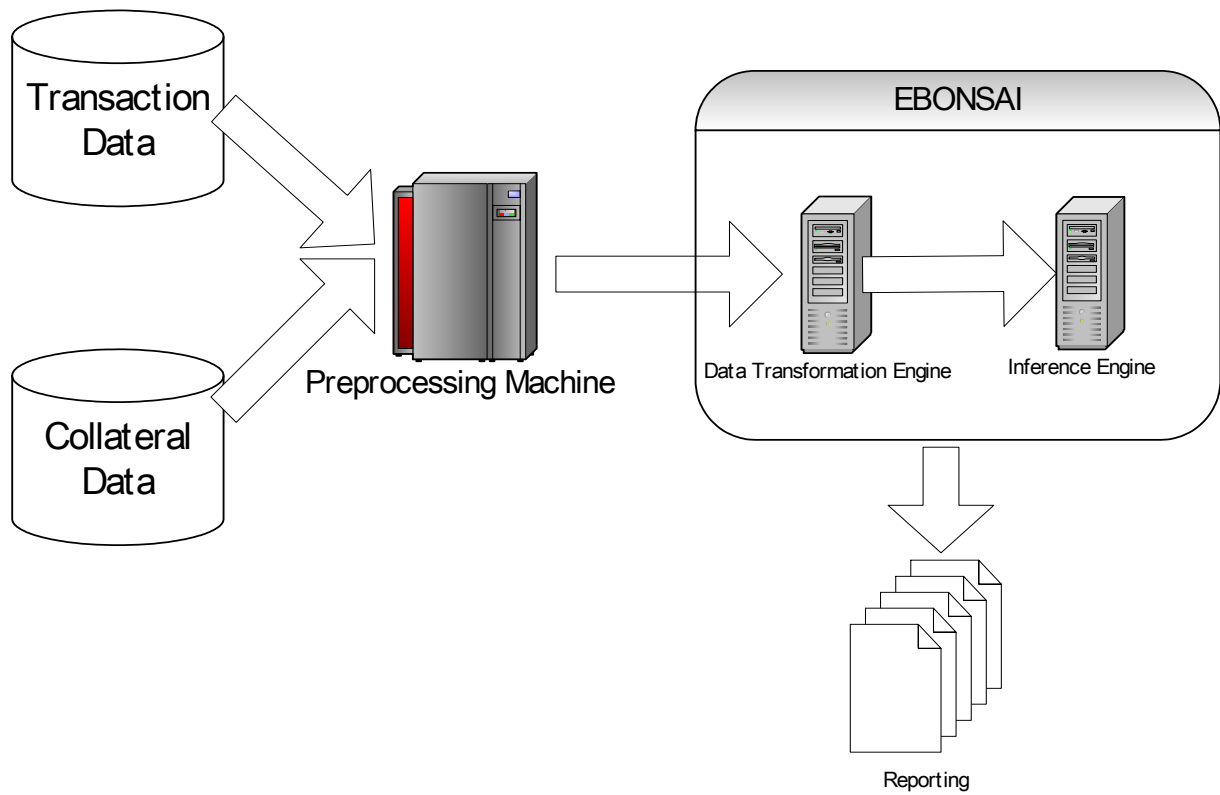


Figure 1. System architecture for a decision system for newly launched products.

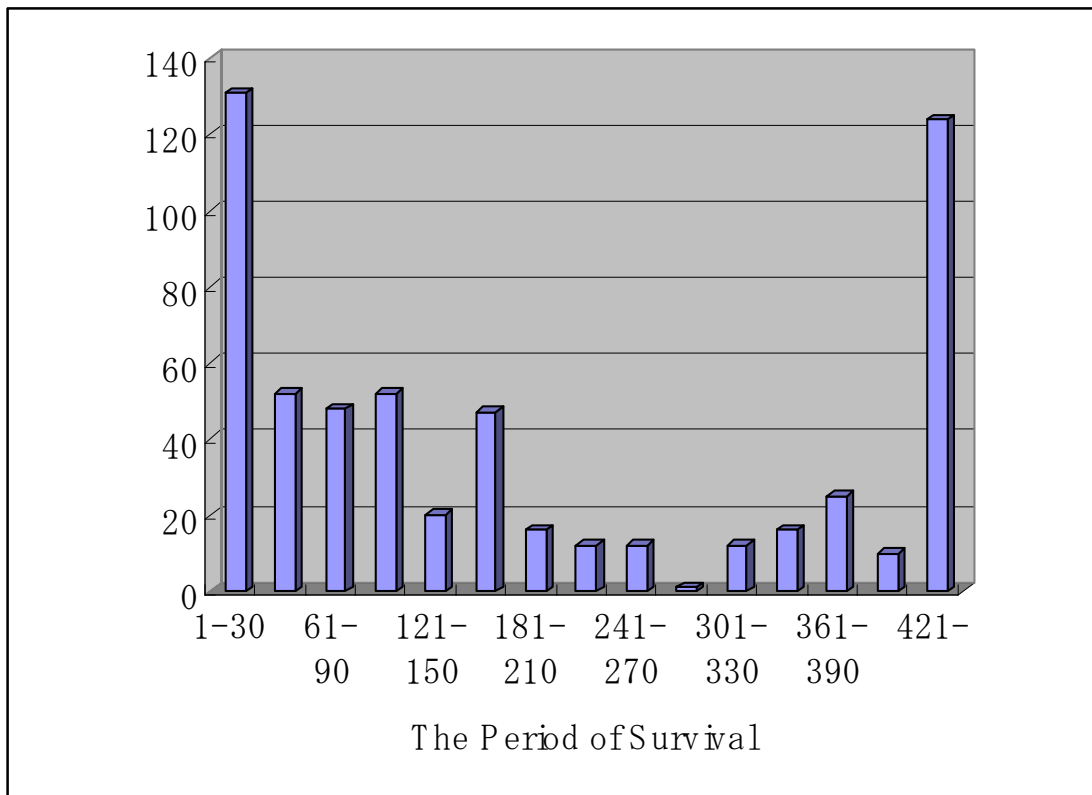


Figure 2: The distribution of survival time of new products. The horizontal axis indicates range of days. The vertical axis indicates counts of brands.

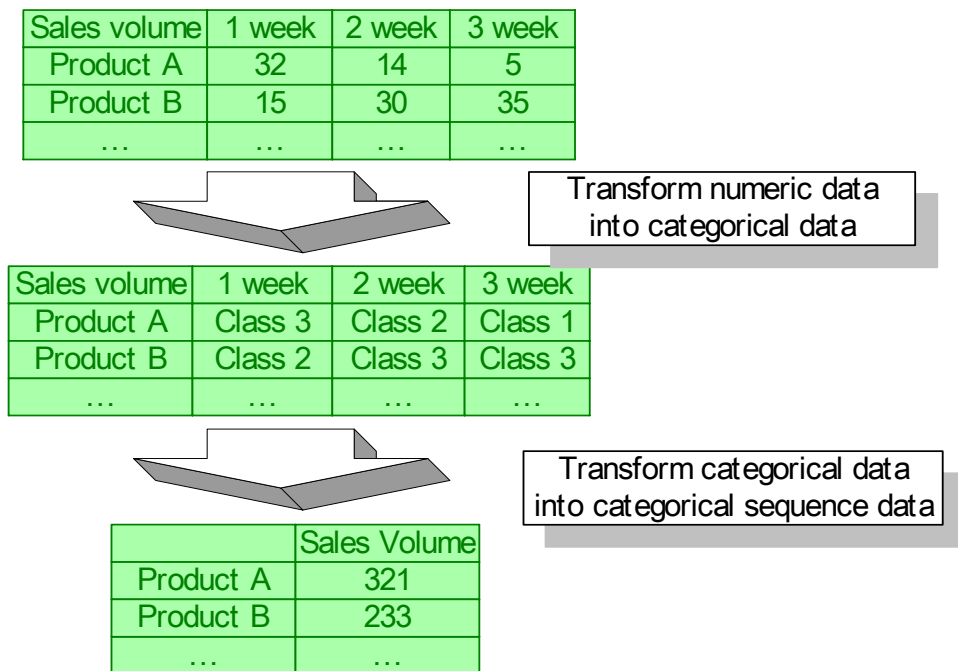


Figure 3. The two-step transformation procedure for input variables.

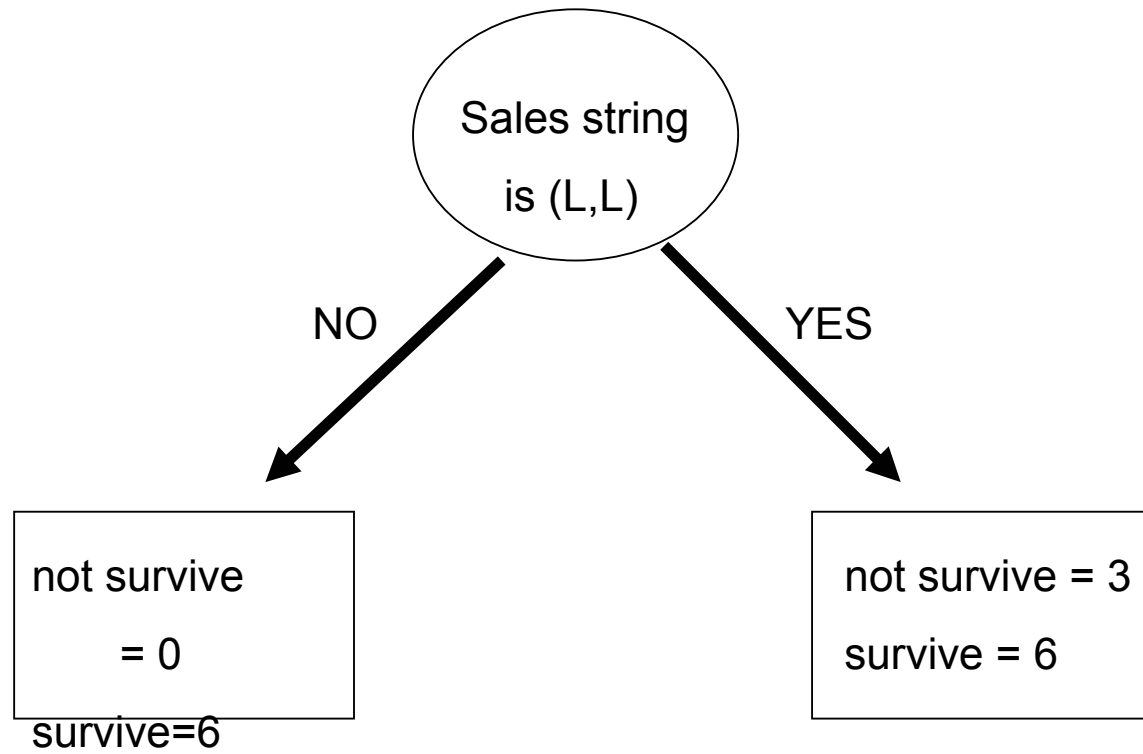


Figure 4a. The tree obtained from EBONSAI applied on the data set in Table 3.

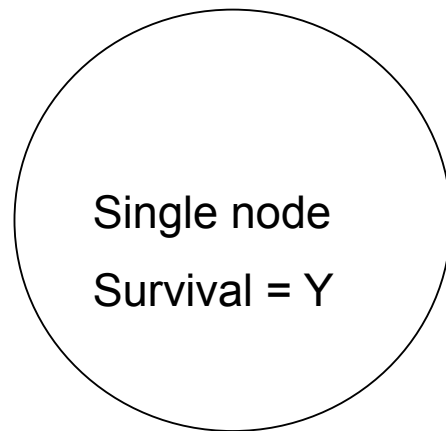


Figure 4b. Single node produced by ID3 applied to data set in Table 3.

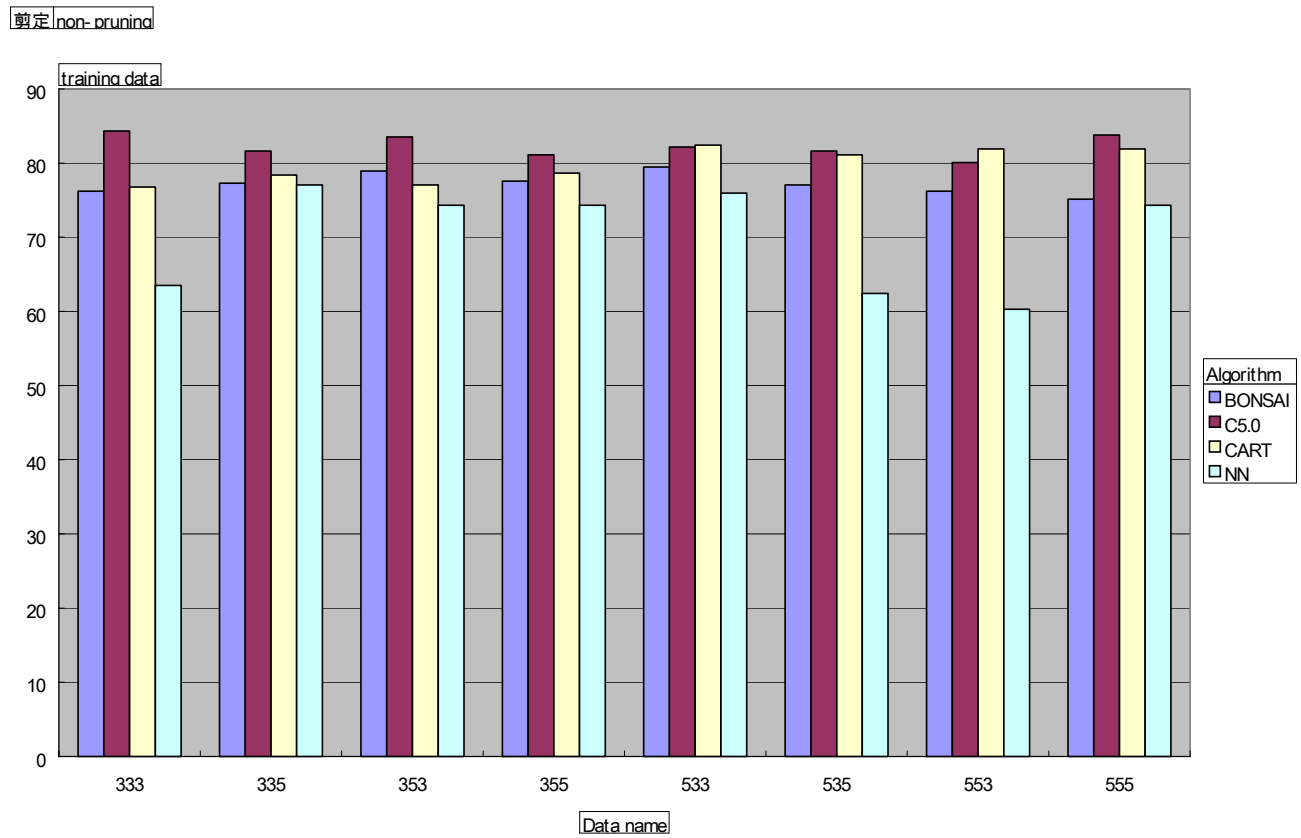


Figure 5a Overall predictive accuracy of EBONSAL, CART and C5.0 on the training set. Vertical axis shows percentage correctly classified.

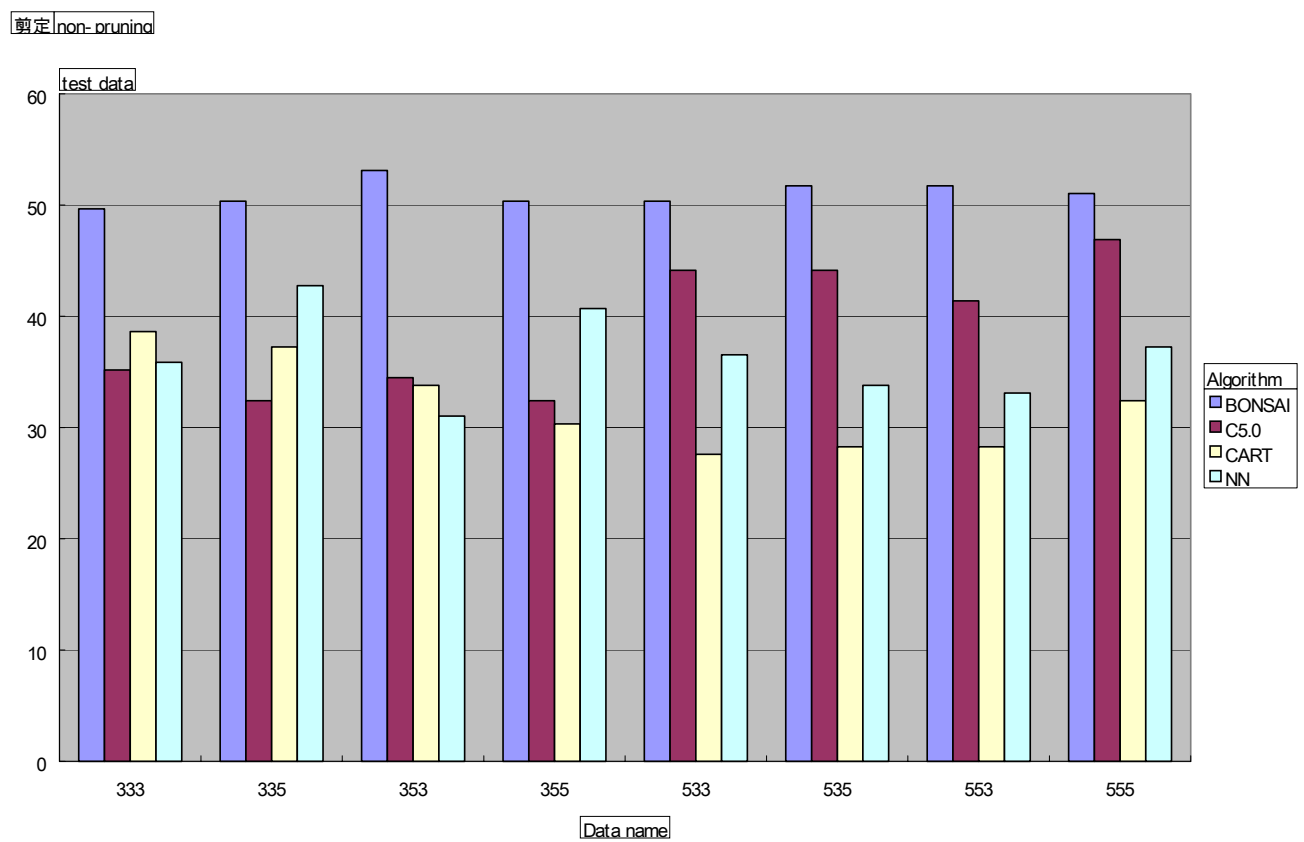


Figure 5b Overall predictive accuracy of EBONSAI, CART and C5.0 on the test set. Vertical axis shows percentage correctly classified.

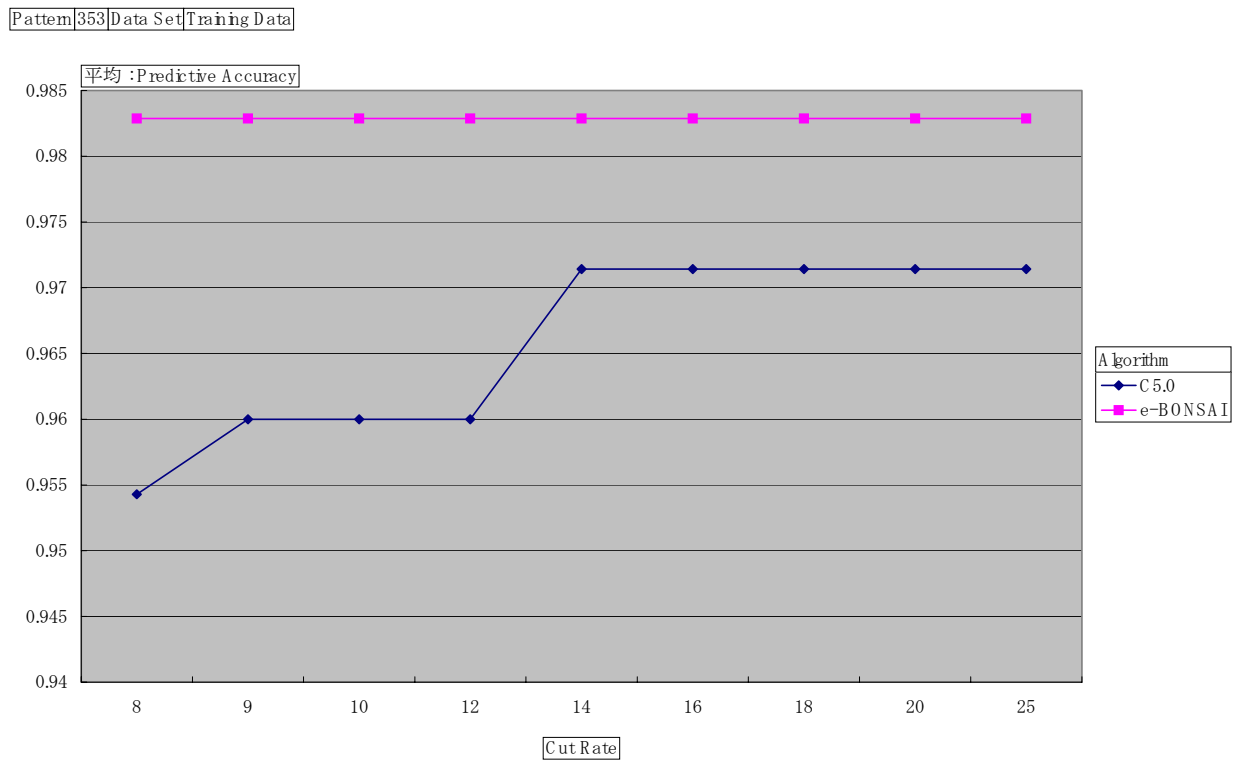


Figure 6 a: Change of prediction accuracy of EBONSAI and C5.0 by varying pruning parameters for the training set. Vertical axis shows percentage correctly classified.

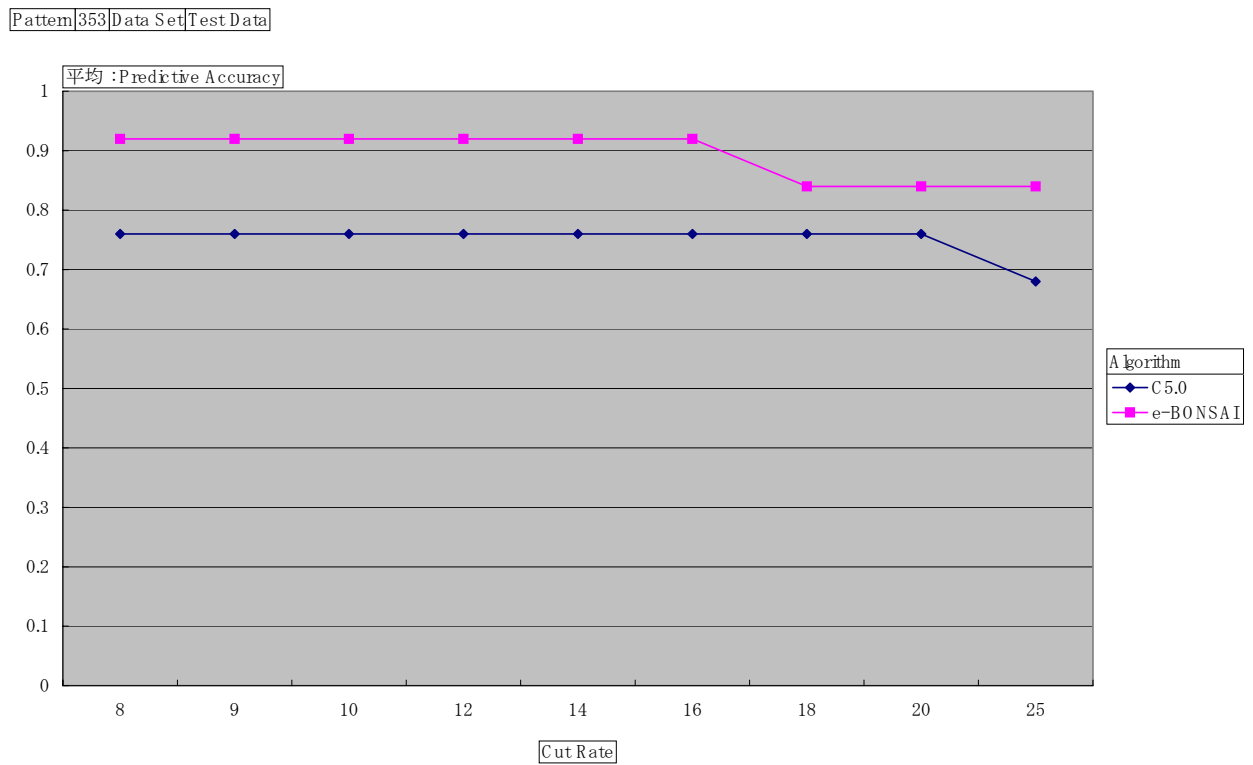


Figure 6 b: Change of prediction accuracy of EBONSAI and C5.0 by varying pruning parameters for the test set. Vertical axis shows percentage correctly classified.