On Implementing a Financial Decision Support System

Sotiris Kotsiantis[†] *, Dimitris Kanellopoulos*[††] *and Vasilis Tampakas*[†]

[†] Technological Educational Institute of Patras, Greece
[††] University of Patras, Department of Electrical & Computer Engineering, Greece

Summary
A successful financial decision support system presents many challenges. Some are encountered over again, and though an individual solution might be system-specific, general principles still apply. This work is twofold aiming to provide a survey of data enhancing techniques, performance improvements, evaluation hints and pitfalls to avoid as well as to exploit them by using a prototype tool that can semi-automate the financial decision support process.

Key words:
Financial predictions, data mining, decision support systems

Introduction

To understand customer needs, preferences, and behaviors, financial institutions such as banks, mortgage lenders, credit card companies, and investment advisors use financial decision support systems. These systems can help companies in the financial sector to uncover hidden trends and explain the patterns that affect every aspect of their overall success.

Financial institutions have large amounts of collected detailed customer data – usually in many disparate databases and in various formats. Only with the recent advances in database technology and data mining techniques have financial institutions acquired the necessary tools to manage their risks using all available information, and explore a wide range of scenarios.

The prediction of user behavior in financial systems can be used in many cases. Predicting client migration, marketing or public relations can save a lot of money and other resources. Another interesting field of prediction is the fraud of credit lines, especially credit card payments (Brause et al., 1999).

What are the characteristics of financial problems that make it difficult to induce robust predictive models? First, the dimensionality of the problem is high. Secondly, relationships among independent and dependent variables are weak and nonlinear. The nonlinearities can be especially pronounced towards the tails of distributions, where a correlation becomes stronger or weaker than it is elsewhere. For example, a common type of nonlinearity in technical analysis is trend reversal, where price trends change direction after a prolonged period. Thirdly, variable interactions can be significant.

For this reason, a successful financial decision support system presents many challenges. Some are encountered over again, and though an individual solution might be system-specific, general principles still apply. Some questions of scientific and practical interest concerning financial decision support systems follow.

- Data preprocessing. Can data transformations that facilitate prediction be identified? In particular, what transformation formulae enhance input data?
- Methods. If prediction is possible, what methods are best at performing it? What methods are best-suited for what data characteristics – could it be said in advance?
- Evaluation. What are the features of sound evaluation procedure, respecting the properties of financial data and the expectations of financial prediction?

The paper addresses many of the questions and remarks on a decision support system development. Using them as guidelines, we have also implemented a prototype tool that might save time, effort and boost results. The presentation follows stages in a decision support system development: data preprocessing, prediction algorithm selection and system evaluation. The paper assumes some familiarity with data mining and financial prediction. As a reference one could use (Kingdon, 1997; Kovalerchuk and Vityaev, 2000).

2. Data Preparation and Data Preprocessing

Firstly, during the problem definition and data preparation task, the data that are relevant to the problem should be collected from the available sources. The data from different resources should be transformed and merged. This step may be not easy and take much time in practice, since the data are usually not in the same format; some data are even not in the electronic format.

The collected cases may have definite classes, for example, the bankrupt and non-bankrupt firms. In other situations, the cases have uncertain classes, for example, it should be decided whether the bad customers are defined as those that are behind in payment for 90 days or 30 days. Sometimes, there are not available past cases with known payment behaviors. The collected cases are analyzed by experts manually and divided into different risk classes.

Manuscript received January, 2006.
Manuscript revised January, 2006.

The tasks in data selection consist of several steps which are illustrated in Figure 1

Since even the best predictor will fail on bad data, data quality and preparation is crucial. Moreover, since a predictor can exploit only certain data features, it is important to detect which data preprocessing/presentation works best (Walczak, 2001). It would be nice if a single sequence of data pre-processing algorithms had the best performance for each data set but this is not the case. Thus, we present the most well known algorithms for each step of data pre-processing so that one achieves the best performance for their data set.

What can be wrong with the data? There is a hierarchy of problems that are encountered:

- Impossible values have been input
- Unlikely values have been input
- Inconsistent values have been input
- No values have been input

Impossible values should be checked for by the data handling software, ideally at the point of input so that they can be re-entered. These errors are generally straightforward like negative prices when positive ones are expected. If correct values cannot be entered, the observation needs to be moved up the hierarchy to the missing value category.

Variable-by-variable data cleaning is straightforward filter approach for unlikely values (those values that are suspicious due to their relationship to a specific probability distribution, say a normal distribution with a mean of 5, a standard deviation of 3, and a suspicious value of 10). Table 1 shows examples of how this metadata can help on detecting a number of possible data quality problems. Moreover, a number of authors focused on the problem of duplicate instance identification and elimination, e.g., (Hernandez & Stolfo, 1998).

Inconsistent values represent a more sophisticated error. An inlier is a data value that lies in the interior of a statistical distribution and is in error. Because inliers are difficult to distinguish from good data values they are sometimes difficult to find and correct. Multivariate data cleaning is more difficult, but is an essential step in a complete analysis (Rocke and Woodruff, 1996). Examples are the distance based outlier detection algorithm RT (Knorr & Ng, 1997) and the density based outliers LOF (Breunig et al., 2000).

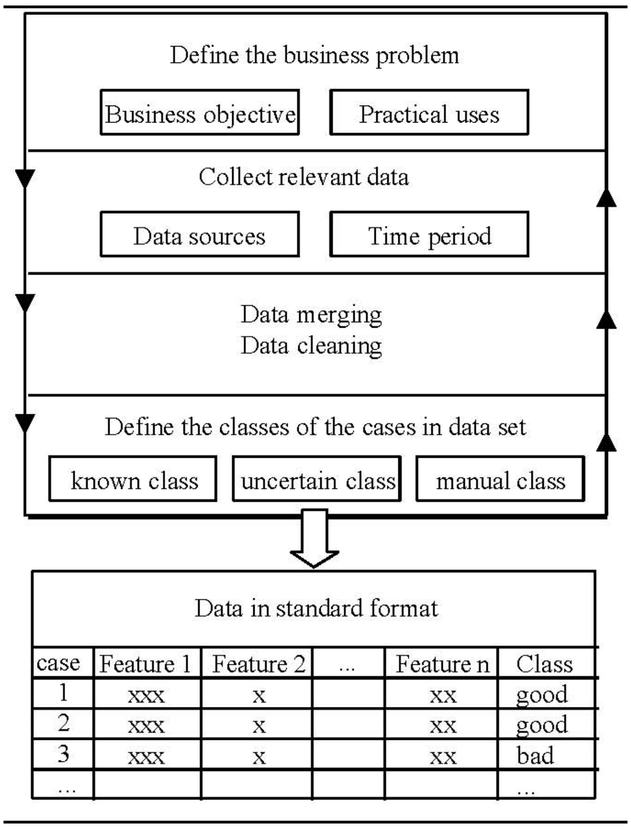


Figure 1. Problem definition and data preparation

Problems	*Metadata*	*Examples/Heuristics*
Illegal values	cardinality	e.g., cardinality (gender) 2 indicates problem
	max, min	max, min should not be outside of permissible range
	variance, deviation	variance, deviation of statistical values should not be higher than threshold
Misspellings	feature values	sorting on values often brings misspelled values next to correct values

Table 1. Examples for the use of variable-by-variable data cleaning

While the focus above has been on analytical methods, the use of visualization can often be a powerful tool. It is particularly good at picking out bad values that are occurring in a regular pattern. For example, simple surface plots will reveal holes or spikes.

A word of caution is needed at this point. First, while automatic methods can detect unusual values that cannot distinguish between values that are unlikely but true and those that are just plain wrong. It may be that someone is satisfied by removing all unlikely values because they are difficult to model but in doing so useful, if awkward, information may be missed. Second, any form of automatic data cleaning will have an effect on the results of any subsequent modeling. In general it is hoped that the cleaning will enhance the results, but it is possible that the cleaning may occasionally distort the results. The effects of data cleaning on the whole process needs to be examined and should not be treated in isolation.

Incomplete data is an unavoidable problem in dealing with most of the real world data sources. The topic has been discussed and analyzed by several researchers (Bruha & Franek, 1996; Liu et al., 1997).Analogically with the case, the expert has to choose from a number of methods for handling missing data (Lakshminarayan et al., 1999):

- *Method of Ignoring Instances with Unknown Feature Values*.
- *Most Common Feature Value:* The value of the feature that occurs most often is selected to be the value for all the unknown values of the feature.
- *Concept Most Common Feature Value:* This time the value of the feature, which occurs the most common within the same class is selected to be the value for all the unknown values of the feature.
- *Mean substitution:* Substitute a feature’s mean value computed from available cases to fill in missing data values on the remaining cases. A smarter solution than using the “general” feature mean is to use the feature mean for all samples belonging to the same class to fill in the missing value.
- *Regression or classification methods:* Develop a regression or classification model based on complete case data for a given feature, treating it as the outcome and using all other relevant features as predictors.
- *Hot deck imputation:* Identify the most similar case to the case with a missing value and substitute the most similar case’s *Y* value for the missing case’s *Y* value.
- *Method of Treating Missing Feature Values as Special Values:* Treating “unknown” itself as a new value for the features that contain missing values.

Given a wide range of possible methods for both error detection and imputation, how can you compare them? One approach is to start with the data set one has, and then perturb the data adding odd values to replace any missing values, and then apply the different methods that one is considering. The results can be evaluated using suitable criterion such as those suggested in (Chambers, 2001).

There are a number of issues that we must also take into account:

- Discretization
- Feature selection
- Instance selection

Discretization can significantly reduce the number of possible values of the continuous feature since large number of possible feature values contributes to slow and ineffective process of data mining learning. The simplest discretization method is an unsupervised direct method named *equal size discretization*. It calculates the maximum and the minimum for the feature that is being discretized and partitions the range observed into *k* equal sized intervals. *Equal frequency* is another unsupervised direct method. It counts the number of values we have from the feature that we are trying to discretize and partitions it into intervals containing the same number of instances. *Entropy* is a more sophisticated supervised incremental top down method described in (Elomaa and Rousu, 1999). Entropy discretization recursively selects the cut-points minimizing entropy until a stopping criterion based on the Minimum Description Length criterion ends the recursion.

Feature subset selection is the process of identifying and removing as much irrelevant and redundant features as possible (Liu and Motoda, 1998). This reduces the dimensionality of the data and enables data mining algorithms to operate faster and more effectively.

Moreover, the problem of *feature interaction* can be addressed by constructing new features from the basic feature set (Markovitch and Rosenstein, 2002). This technique is called *feature construction/transformation*. The new generated features may lead to the creation of more concise and accurate classifiers. In addition, the discovery of meaningful features contributes to better comprehensibility of the produced classifier, and better understanding of the learned concept.

For instance, suppose we try to predict whether the shares of a company will go up or down in the financial market, based on a set of predictor features that includes both the company’s total income and the company’s total expenditure in the last 12 months. Most rule induction algorithms are not capable of discovering rules of the form:

IF (Income > Expenditure) AND . . . THEN (Shares = up)
IF (Income < Expenditure) AND . . . THEN (Shares = down) ,

because those algorithms do not have the autonomy to create feature-feature (rather tha feature-value) rule conditions. Clearly, this limitation would be removed if we construct the new boolea feature “Income > Expenditure?”.

Instance selection isn’t only used to handle noise but for coping with the infeasibility of learning from very large data sets. Instance selection in this case is an optimization problem that attempts to maintain the mining quality while minimizing the sample size (Liu and Metoda, 2001). It reduces data and enables a data mining algorithm to function and work effectively with huge data. There is a variety of procedures for sampling instances from a large data set. The most well known are (Reinartz, 2002):

- *Random sampling* that selects a subset of instances randomly.
- *Stratified sampling* that is applicable when the class values are not uniformly distributed in the training sets. Instances of the minority class(es) are selected with a greater frequency in order to even out the distribution.

It would be nice if a single data mining algorithm had the best performance for each data set but this is not the case. Thus, in the following section, we present the most well known data mining algorithms so that one achieves the best performance for their data set.

3. Prediction Algorithms

In a standard classification problem the input instances are associated with one of *k* unordered sets of labels denoting the class membership. Since the target values are unordered, the metric distance between the prediction and the correct output is the non-metric 0-1 indicator function. In a standard regression problem target values range over the real numbers therefore the loss function can take into account the full metric structure.

The problem of predicting stock market returns or exchange rates at time t+1 can be cast as either a regression or classification problem. Whereas the regression problem for exchange rate data involves modeling the actual exchange rate, the classification problem involves predicting whether the exchange rate has increased or decreased.

As an example of a classification, consider the problem of trading a future of stock A at price B on date C by using a learning algorithm. Firstly, the historical data is prepared. At each time step, data are classified into one of two categories according to whether it was profitable to buy or sell stock A at price B on date C. Having fitted a model with this historical data, the model can be used to predict a profitable position at time t+1 (e.g., the next day or week). At the end of each time step the model is updated to include the new historical data.

Murthy (1998) provides an overview of existing work in decision trees, and a taste of their usefulness, to the newcomers. In rule induction systems, a decision rule is defined as a sequence of Boolean clauses linked by logical AND operators that together imply membership in a particular class (Furnkranz, 1999). Model rules are the counterpart of rule learners for regression tasks.

Artificial Neural Networks (ANN) depend upon three fundamental aspects, the input and activation functions of the unit, the network architecture and the weight on each of the input connections (Mitchell, 1997).

A Bayesian Network (BN) is a graphical model for probabilistic relationships among a set of variables (Jensen, 1996). Naive Bayes (NB) classifier (Domingos and Pazzani, 1997) is the simplest form of Bayesian network.

The k-Nearest Neighbour (kNN) approach is based on the principal that the instances within a data set will generally exist in close proximity with other instances that have similar properties (Aha, 1997). Locally weighted linear regression (LWR) is a combination of instance-based learning and linear regression.

SVM technique revolves around the notion of a 'margin', either side of a hyperplane that separates two data classes. Maximising the margin and thereby creating the largest possible distance between the separating hyperplane and the instances on either side of it, is proven to reduce an upper bound on the expected generalisation error. An excellent survey of SVMs can be found in (Burges, 1998). The SVM can be extended to regression estimation by introducing an insensitive loss function (Shevade, 2000).

4. Evaluation

Actually, the most well-known classifier criterion is its accuracy. A confusion matrix shows the type of classification errors a classifier makes. Table 2 represents a confusion matrix for the two-class case – the extension to the multi-class problem is straightforward. The breakdown of a confusion matrix is as follows: *a* is the number of positive instances correctly classified, *b* is the number of positive instances misclassified as negative, *c* is the number of negative instances misclassified as positive, *d* is the number of negative instances correctly classified.

Hypothesis (prediction)		
+	-	Actual Class
a	*b*	+
c	*d*	–

Table 2. A confusion matrix

Accuracy (denoted as *acc*) is commonly defined over all the classification errors that are made and it is calculated as: $acc = (a+d)/(a+b+c+d)$. The error rate can be reversely calculated as $error_{rate}$=1-*acc*.

For the regression methods, there isn't only one regressor's criterion. Table 3 represents the most well known. Fortunately, it turns out that in most practical situations the best regression method is still the best no matter which error measure is used.

Mean absolute error	$\frac{\lvert p_1 - a_1 \rvert + \mathrm{K} + \lvert p_n - a_n \rvert}{n}$
Root mean squared error	$\sqrt{\frac{(p_1 - a_1)^2 + \mathrm{K} + (p_1 - a_1)^2}{n}}$
Relative absolute error	$\frac{\lvert p_1 - a_1 \rvert + \mathrm{K} + \lvert p_n - a_n \rvert}{\lvert a_1 - \bar{a} \rvert + \mathrm{K} + \lvert a_n - \bar{a} \rvert}$
Root relative squared error	$\sqrt{\frac{(p_1 - a_1)^2 + \mathrm{K} + (p_1 - a_1)^2}{(a_1 - \bar{a})^2 + \mathrm{K} + (a_1 - \bar{a})^2}}$

Table 3. Regressor criteria (p : predicted values, a : actual values, $\bar{a} = \left(\sum_i a_i \right) / n$

5. Methodology

One can imply from the previous sections the process of applying data mining algorithms to a real-world financial problem. It is briefly described in Figure 2. The first step is the collection of the data set. If a domain expert is available, then he/she could suggest which fields (attributes, features) are the most informative. If not, then the simplest method is a "brute-force", which indicates the measuring of everything available and only hopes that the right (informative, relevant) features are among them. However, a data set collected by the "brute-force" method is not directly suitable for induction. It comprises in most cases noise and missing values, therefore it needs pre-processing. The choice of which specific data mining algorithm to use is a critical step. Once preliminary testing is judged to be satisfactory, the classifier/regressor is available for routine use. If the testing is not satisfactory, we must return to a previous stage. The earlier stage we return to, the more time we spent but the result may be better.

As we have already mentioned, the pre-processing step is necessary to resolve several types of problems including noisy data, redundancy data, missing data values, etc. All the data mining algorithms rely heavily on the product of this stage, which is the final training set. By selecting relevant instances, experts can usually remove irrelevant ones as well as noise and/or redundant data. The high quality data will lead to high quality results and reduced costs for data mining. In addition, when a data set is too huge, it may not be possible to run a data mining algorithm. In this case, instance selection reduces data and enables a data mining algorithm to function and work effectively with huge data.

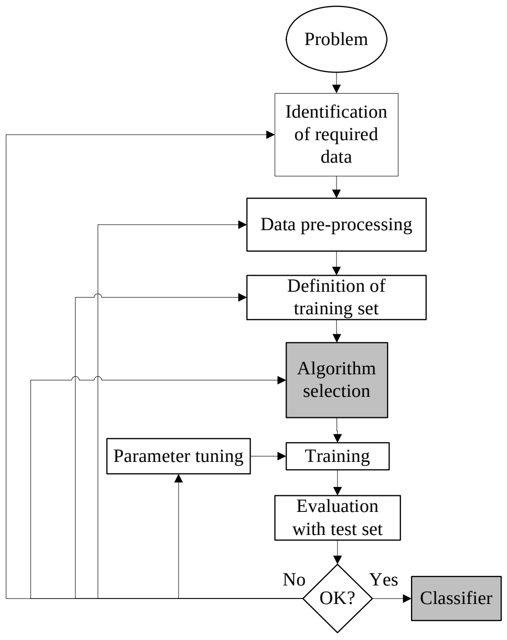


Figure 2. The process of financial predictions

Most of the existing decision tree, rule based and Bayesian data mining algorithms are able to extract knowledge from data set that store discrete features. If the features are continuous, these algorithms can be integrated with a discretization algorithm that transforms them into discrete features.

Moreover, normalization is a "scaling down" transformation of the features. Within a feature there is often a large difference between the maximum and minimum values, e.g. 0.01 and 1000. When normalization is performed the value magnitudes and scaled to appreciably low values. This is important for neural network and k-NN algorithms.

The choice between feature selection and feature construction depends on the application domain and the specific training data, which are available. Feature selection leads to savings in measurements cost since some of the features are discarded and the selected features retain their original physical interpretation. In addition, the retained features may be important for understanding the physical process that generates the patterns. On the other hand, transformed features generated by feature construction may provide a better discriminative ability than the best subset of given features, but these new features may not have a clear physical meaning.

Finally, the suggested data pre-processing methodology is summarized in Figure 3.

Generally, statistical methods (e.g. SVM, neural networks) tend to perform much better over multi-dimensions and continuous features. By contrast, rule-based systems (e.g.

Decision trees, rule learners) tend to perform better in discrete/categorical features. For neural network models and SVM, large sample size is required in order to achieve its maximum prediction accuracy whereas NB may need a relatively small dataset.

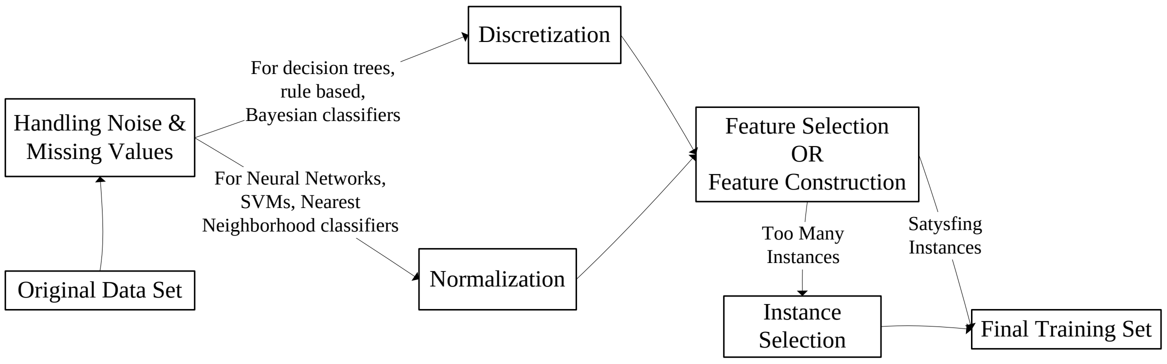

Discretization
For decision trees, rule based, Bayesian classifiers
Handling Noise & Missing Values
For Neural Networks, SVMs, Nearest Neighborhood classifiers
Original Data Set
Normalization
Feature Selection OR Feature Construction
Too Many Instances
Satysfing Instances
Instance Selection
Final Training Set

Figure 3. Data pre-processing steps

Decision trees cannot perform well with problems that require diagonal partitioning. The division of the instance space is orthogonal to the axis of one variable and parallel to all other axes. Therefore, the resulting regions are all hyperrectangles. The ANNs and the SVMs perform well when multicollinearity is present and a nonlinear relationship exists between the input and the output features.

Although training time varies according to the nature of the application task and dataset, specialists generally agree on a partial ordering of the major classes of data mining algorithms. For instance, kNN require zero training time because the training instance is simply stored. Naive Bayes method also train very quickly since they require only a single pass on the data either to count frequencies (for discrete variables) or to compute the normal probability density function (for continuous variables under normality assumptions). Univariate decision trees are also reputed to be quite fast—at any rate, several orders of magnitude faster than neural networks and SVMs.

Naive Bayes requires little storage space during both the training and classification stages: the strict minimum is the memory needed to store the prior and conditional probabilities. The basic kNN algorithm uses much storage space for the training phase, and execution space is at least as big as training space. On the contrary, for all non lazy learners (i.e. kNN), execution space is usually much smaller than training space, since the resulting classifier is usually highly condensed summary of the data.

Moreover, kNN is generally considered intolerant of noise; its similarity measures can be easily distorted by errors in feature values, thus leading it to misclassify a new instance on the basis of the wrong nearest neighbors. Contrary to kNN and set-covering rule learners, most decision trees are considered resistant to noise because their tree pruning strategies avoid overfitting the data in general and noisy data in particular.

Furthermore, the number of model or runtime parameters to be tuned by the user is an indicator of an algorithm's ease of use. It can help in prior model selection based on the user's priorities and preferences: for a non specialist in data mining, an algorithm with few user-tuned parameters will certainly be more appealing, while a more advanced user might find a large parameter set an opportunity to control the data mining process more closely. As expected, neural networks and SVMs have more parameters than the remaining techniques.

Logic-based algorithms like decision trees and rule inducers are all considered very easy to interpret, whereas neural networks and SVM have notoriously low interpretability. k-NN is also considered to have very poor interpretability because an unstructured collection of training instances is far from readable, especially if there are many of them.

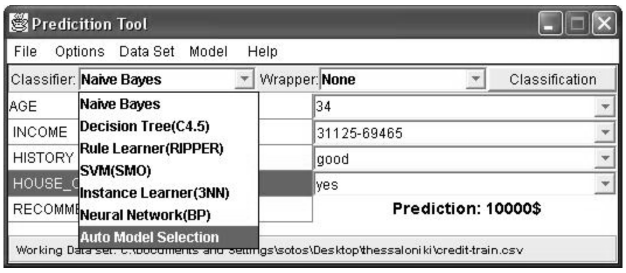
In ordinal classification, the target values are in a finite set (like in classification) but there is an ordering among the elements (like in regression, but unlike classification). A sophisticated approach that enables standard classification algorithms to make use of ordering information in ordinal class features is presented in (Kotsiantis and Pintelas, 2004).

6. Prototype Tool

The above mentioned stages denote general guidelines and thus they do not provide explicitly a path for selecting the most informative features and the most accurate learning algorithm for a given problem. For this reason, we have implemented a prototype tool that can automatically select the most useful features for the given problem as well as the most accurate learning algorithm for the given problem. The tool expects the training set as a spreadsheet in CSV (Comma-Separated Value) file format. The CSV file format is often used to exchange data between disparate applications. The file format, as it is used in Microsoft Excel, has become a pseudo standard throughout the industry, even among non-Microsoft platforms. The tool assumes that the first row of the CSV file is used for the names of the features. There is not any restriction in features' order. However, the class feature must be in the last column.

After opening the data set that characterizes the problem for which the user wants to take the prediction, the tool automatically uses the corresponding features for training the selected algorithm. Naive Bayes (Domingos & Pazzani, 1997) was used as the representative of Bayesian Networks. The most commonly used C4.5 algorithm (Quinlan, 1993) was used as the representative of the decision trees in our tool. The most well known learning algorithm which is used to estimate the values of the weights of a neural network – the Back Propagation (BP) algorithm (Mitchell, 1997) – was the representative of the ANNs in our tool. The RIPPER algorithm (Cohen, 1995) was the representative of the rule-learning techniques in our tool because it is one of the most often used methods that produce classification rules. The 3-NN algorithm combines resistance to noise with less time for classification than using a larger k for kNN, which was used in our tool (Aha, 1997). Finally, the Sequential Minimal Optimization (or SMO) algorithm was the representative of the SVMs in our tool as one of the fastest methods to train SVMs (Platt, 1999).

A screen shot of the decision support tool is presented in Figure 4. The user can let the tool find the most accurate algorithm for the specific data set via 'Auto model selection'. Cross validation is a model evaluation method that is better than residuals. The problem with residual evaluations is that they do not give an indication of how well the learner will do when it is asked to make new predictions for data it has not already seen. One way to overcome this problem is to not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on ``new" data. This is the basic idea for a whole class of model evaluation methods called cross validation.



Classifier: Naive Bayes Wrapper: None

AGE 34
INCOME 31125-69465
HISTORY good
HOUSE_C yes
RECOMME

Prediction: 10000$

Working Data set: C:\Documents and Settings\sotos\Desktop\thessaloniki\credit-train.csv

Figure 4. Prototype tool

The used methodology for 'Auto model selection' is the following four steps strategy (3-cross validation):

- The data set is divided at random into three equal parts.
- Two of these parts are used for training the algorithms and the remaining data is the testing set. This process is repeated three times
- The results of three tests are averaged and the algorithm that achieves the highest accuracy is selected.
- The selected algorithm then executes on the full training set to produce the prediction model.

It must be mentioned that after training the learning algorithm, the user is able to see the produced classifier.

Recently, in the area of data mining the concept of combining learning algorithms is a new direction for the improvement of the performance of individual algorithms. (Dietterich, 2001). Despite the obvious advantage of ensembles' accuracy, there are at least two weaknesses: (1) increased computation, and (2) decreased comprehensibility. For this reason, we used another way of improving the classification accuracy that at the same time maintains the comprehensibility. We included feature selection techniques in the prototype tool that identifies and removes irrelevant features for improving the accuracy.

Generally, there are two models of feature subset selection: the filter and the wrapper model. In the filter model, the features are filtered independently of the induction algorithm (Liu & Motoda, 1998). Feature wrappers often achieve better results than filters due to the fact that they are tuned to the specific interaction between an induction algorithm and its training data. For this reason, we included wrapper techniques in the prototype tool.

There are several wrapper selection algorithms that try to evaluate the different subsets of the features on the learning algorithm and keep the subsets that perform best. The simplest method is forward selection (FS) (Liu & Motoda, 1998). It starts with the empty set and greedily adds features one at a time. At each step FS adds the feature that, when added to the current set, yields the learned structure that generalizes best. Once a feature is added FS cannot later remove it. Backward stepwise selection (BS) starts with all features in the feature set and greedily removes them one at a time, too. Like forward selection, backward selection removes at each step the feature whose removal yields a set of features that yields best generalization. Also like FS, once BS removes a feature, it cannot later add it back to the set.

Sequential forward floating selection (SFFS) and sequential backward floating selection (SBFS) are characterized by the changing number of features included or eliminated at different stages of the procedure. A similar way is followed by the Best First search. The Best First search starts with an empty set of features and generates all possible single feature expansions (Kohavi and John, 1997). The subset with the highest evaluation is chosen and is expanded in the same manner by adding single features. If expanding a subset results in no improvement, the search drops back to the next best unexpanded subset and continues from there. Given enough time a Best First search will explore the entire search space, so it is common to limit the number of subsets expanded that result in no improvement. The best subset found is returned when the search terminates. The Best First search can be combined with forward or backward selection. Another way is to start the search from a randomly selected subset (i.e. Random Generation) and add or delete a feature at random. A more informed random feature selection is carried out with the use of genetic algorithms (Witten & Frank, 2000).

However, none of the feature selection algorithms consistently exhibits superior performance in all data sets. For this reason, the user can let the tool find the most suitable feature selection algorithm for the specific learning algorithm and data set via 'Auto model selection'.

The used methodology for 'Auto model selection' is the following four steps strategy:

- The data set is divided at random into three equal parts.
- Two of these parts are used as input to the wrapper feature selection algorithms and the remaining data is the testing set. This process is repeated three times
- The results of three tests are averaged and the feature selection algorithm that achieves the highest accuracy for the specific learning algorithm is selected.

Of course, the process of feature selection takes some extra time to complete. The tool can predict the class of either a single instance or an entire set of instances (batch of instances). It must be mentioned that for batch of instances the user must import an Excel cvs file with all the instances he/she wants to have predictions. Moreover, the implemented tool can present useful information about the imported data set such as the presence or not of missing feature values, the frequency of each feature value etc. In the next sub-section, we refer to a use case of the prototype tool.

6.1 Use Case of the Prototype Tool

For the purpose of the use case, we used a credit rating dataset from the UCI Repository (Blake & Merz, 1998). Each case represents an application for credit card facilities described by eight discrete and six continuous features, with two decision classes (Accept / Reject). In the UCI Repository, the original feature names have been changed to meaningless symbols (A1 – A14) with the purpose of protecting the confidentiality of the data. However, the real names of the features are available at the site of Rulequest Research [http://www.rulequest.com/see5-examples.html]. The database features are shown in *Table 4*. The original names of the features (provided by the Rulequest Research site) are given in parentheses. The *Domain* column shows the set or range of possible values for each feature. In the *Type* column, we make a distinction between discrete (nominal) and continuously valued features.

The following rule was produced by the implemented tool:

If (A9=true) then Accept else Reject (accuracy 85.51 %)

In the next section, we refer to the applications of financial decision support systems

Feature	**Domain**	**Type**
A1 (Sex)	0, 1	Nominal
A2 (Age)	13.75 - 80.25	Continuous
A3 (Mean time at addresses)	0 - 28	Continuous
A4 (Home status)	1, 2, 3	Nominal
A5 (Current occupation)	1 - 14	Nominal
A6 (Current job status)	1 - 9	Nominal
A7 (Mean time with employers)	0 - 28.5	Continuous
A8 (Other investments)	0, 1	Nominal
A9 (Bank account)	0, 1	Nominal
A10 (Time with bank)	0 - 67	Continuous
A11 (Liability reference)	0, 1	Nominal
A12 (Account reference)	1, 2, 3	Nominal
A13 (Monthly housing expense)	0 - 2000	Continuous
A14 (Savings account balance)	1 - 100001	Continuous
Class (Reject / Accept)	0, 1	Nominal

Table 4. Credit Approval Dataset – List of Features

6. Applications of Financial Decision Support Systems

Although rating agencies and many institutional writers emphasize the importance of analysts' subjective judgment in determining credit ratings, many researchers have obtained promising results on credit rating prediction applying different data mining methods (Maher and Sen, 1997). The overall objective of credit rating prediction is to build models that can extract knowledge of credit risk evaluation from past observations and to apply it to evaluate credit risk of companies with much broader scope. Prediction of corporate bankruptcy is a phenomenon of increasing interest to investors/creditors, borrowing firms, and governments alike. Timely identification of firms' impending failure is indeed desirable. Most of the bankruptcy prediction studies have used financial ratios (including the ones measuring liquidity, solvency, leverage, profitability, asset composition, firm size, growth etc.) to predict failure in firms (Jo et al., 1997). Other variables of interest include information on macroeconomic, industry specific, location or spatial, and firm specific variables (Pompe and Feelders, 1997).

Moreover, other researchers were interested in the relationship between a number of macro/microeconomic indicators of countries/companies and different economic/financial performance classifications (Costea and Eklund, 2003). Instead of analyzing fundamental information about companies, the technical approach tries to identify turning points, momentum, levels, and directions of an investment instrument, using tools such as charting, relative strength index, moving averages, on balance volume, momentum and rate of change, breadth advance decline indicator, directional movement indicator, and detrended price oscillator. There are divergent opinions about what other trends in the macroeconomic, political, monetary, and societal sentiment spheres should be incorporated in the analysis (Dropsy, 1996).

Moreover, risk assessment is complex and difficult especially because the internal auditor is faced with large amounts of both qualitative and quantitative data. Consequently, the internal auditor's ability to use emerging technologies with data mining capabilities has the potential to enhance audit quality and performance (Ramamoorti and Traver, 1998).

6. Conclusions

Financial markets generate large volumes of data. Analyzing these data to reveal valuable information and making use of the information in decision making, present great opportunities and grand challenges for financial prediction systems. The rewards for finding valuable patterns are potentially huge, but so are the difficulties.

Financial prediction presents challenges encountered over again. This paper highlights some of the problems and solutions. A predictor development demands excessive experimentation: with data preprocessing and selection, the prediction algorithm(s), evaluation and tuning – to benefit from the minute gains, but not fall into over-fitting. For this reason, a prototype tool has also been implemented that can semi-automate the financial decision support process. Of course, more algorithms must be included in the prototype tool in the near future, especially for the data preprocessing phase.

Appendix

The tool is available in the web page:
http://www.math.upatras.gr/~esdlab/PrototypeTool/

References

Aha, D, (1997). Lazy Learning. Dordrecht: Kluwer Academic Publishers.

Blake, C.L. & Merz, C.J. (1998), UCI Repository of machine learning databases. Irvine, CA: University of California, Department of Information and Computer Science. [http://www.ics.uci.edu/~mlearn/MLRepository.html]

Brause R. , Langsdorf T., Hepp M. (1999), Neural Data Mining for Credit Card Fraud Detection, 11th IEEE International Conference on Tools with Artificial Intelligence November 08 - 10, Chicago, Illinois.

Breunig M. M., Kriegel H.-P., Ng R. T., Sander J. (2000): 'LOF: Identifying Density-Based Local Outliers', Proc. ACM SIGMOD Int. Conf. On Management of Data (SIGMOD 2000), Dallas, TX, 2000, pp. 93-104.

Bruha and F. Franek (1996), Comparison of various routines for unknown attribute value processing: covering paradigm. International Journal of Pattern Recognition and Artificial Intelligence, 10, 8, 939-955

Burges, C.J.C. (1998), A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2(2):1-47.

Chambers R. (2001), Evaluation Criteria for Statistical Editing and Imputation, National Statistics Methodological Series No. 28, HMSO, Available at http://www.cs.york.ac.uk/euredit/

Cohen, W. (1995), Fast Effective Rule Induction, Proceeding of International Conference on Machine Learning pp. 115-123.

Costea, A., Eklund (2003), T., A Two-Level Approach to Making Class Predictions, 36th Annual Hawaii International Conference on System Sciences (HICSS'03) - Track 3 January 06 - 09.

Dietterich T.G. (2001), Ensemble methods in machine learning. In J. Kittler, F. Roli, eds., Multiple Classifier Systems (LNCS Vol. 1857), Springer, pp. 1–15.

Domingos P. & Pazzani M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29, 103-130.

Dropsy V. (1996), Do macroeconomic factors help in predicting international equity risk premia? Journal of Applied Business Research 12 (3) 120–132.

Elomaa, T. & Rousu, J. (1999). General and Efficient Multisplitting of Numerical Attributes. Machine Learning 36, 201–244.

Furnkranz, J. (1999), "Separate-and-Conquer Rule Learning", Artificial Intelligence Review, Vol. 13, pp. 3-54.

Jensen, F. (1996). An Introduction to Bayesian Networks. Springer.

Jo, H., Han, I., Lee, H. (1997). Bankruptcy Prediction using Case-Based Reasoning, Neural Networks, and Discriminant Analysis. Expert Systems With Applications 13 (2), 97-108.
Hernandez, M.A.; Stolfo, S.J. (1998): Real-World Data is Dirty: Data Cleansing and the Merge/Purge Problem. Data Mining and Knowledge Discovery 2(1):9-37.
Kingdon, J. (1997). Intelligent systems and financial forecasting. Springer.
Knorr E. M., Ng R. T. (1997): "A Unified Notion of Outliers: Properties and Computation", Proc. 4th Int. Conf. on Knowledge Discovery and Data Mining (KDD'97), Newport Beach, CA, pp. 219-222.
Kohavi R., & John, G. H. (1997), Wrappers for feature subset selection. Artificial Intelligence vol. 97, pp. 273-324.
Kotsiantis S., Pintelas P. (2004), A Cost Sensitive Technique for Ordinal Classification Problems, Lecture Notes in Computer Science, Springer-Verlag Vol 3025, pp. 220 – 229.
Kovalerchuk, B., & Vityaev, E. (2000). Data mining in finance: Advances in relational and hybrid methods. Kluwer Academic.
Lakshminarayan K., S. Harp & T. Samad (1999), Imputation of Missing Data in Industrial Databases, Applied Intelligence 11, 259–275, Kluwer Academic Publishers.
Liu, W.Z., White, A.P., and Thompson S.G., Bramer M.A. (1997), Techniques for Dealing with Missing Values in Classification. In IDAf97, Vol.1280 of Lecture notes, 527-536.
Liu, H. & Motoda H. (1998), Feature Extraction, Construction and Selection: A Data Mining Perspective, Kluwer.
Liu, H. and H. Metoda (2001), Instance Selection and Constructive Data Mining, Kluwer, Boston, MA.
Maher J.J., Sen T.K. (1997), Predicting bond ratings using neural networks: a comparison with logistic regression, Intelligent Systems in Accounting, Finance and Management 6: 59– 72.
Markovitch S. & Rosenstein D. (2002), Feature Generation Using General Constructor Functions, Machine Learning, 49, 59–98, Kluwer Academic Publishers.
Mitchell, T. (1997), Machine Learning, McGraw Hill.
Murthy (1998), Automatic Construction of Decision Trees from Data: A Multi-Disciplinary Survey, Data Mining and Knowledge Discovery, 2, 345–389.
Platt, J. (1999), Using sparseness and analytic QP to speed training of support vector machines. In: Kearns, M. S., Solla, S. A. & Cohn D. A. (Eds.), Advances in neural information processing systems 11. MA: MIT Press.
Pompe, P., Feelders, A. (1997). Using Machine Learning, Neural Networks, and Statistics to Predict Corporate Bankruptcy. Microcomputers in Civil Engineering 12, 267-276.
Ramamoorti, S. and Traver, R.O. (1998), Using Neural Networks for Risk Assessment in Internal Auditing: A Feasibility Study, Institute of Internal Auditors' Research Foundation, Altamonte Springs, FL.
Reinartz T. (2002), A Unifying View on Instance Selection, Data Mining and Knowledge Discovery, 6, 191–210, Kluwer Academic Publishers.
Rocke, D. M. and Woodruff, D. L. (1996) "Identification of Outliers in Multivariate Data", Journal of the American Statistical Association, 91, 1047–1061.
Shevade, S. K.,Keerthi, S. S., Bhattacharyya, C.,&Murthy, K. R. K. (2000). Improvements to the SMO algorithms for SVM regression. IEEE Transactions on Neural Networks, 11:5, 1188–1193.
Walczak, S. (2001). An empirical analysis of data requirements for financial forecasting with neural networks, Journal of Management Information Systems Vol. 17 No. 4, pp. 203 – 222
Witten, I. & Frank, E. (2000), Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, Morgan Kaufmann, San Mateo, CA.

Sotiris Kotsintis received a diploma in mathematics, a Master and a Ph.D. degree in computer science from the University of Patras, Greece. His research interests are in the field of data and multimedia mining. He has more than 40 publications to his credit in international journals and conferences.

Dimitris Kanellopoulos received a diploma in electrical engineering and a Ph.D. degree in electrical and computer engineering from the University of Patras, Greece. Since 1990, he was a research assistant in the Department of Electrical and Computer Engineering at the University of Patras and involved in several EU R&D projects (e.g. RACE I and II, ESPRIT). His research interests are in the field of hard real-time multimedia communication protocols, ATM networking, new OSI services and web engineering. He has more than 30 publications to his credit in international journals and conferences in these areas. Dr. Kanellopoulos is a member of the Technical Chamber of Greece.