

## Agricultural databases evaluation with machine learning procedure

Mahyar Amini<sup>1,2</sup> and Ali Rahmani<sup>1</sup>

<sup>1</sup> MahamGostar Research Group, Iran

<sup>2</sup> University Technology Malaysia (UTM), Malaysia

### ABSTRACT

This paper reviews our experience with the application of machine learning techniques to agricultural databases. We have designed and implemented a machine learning workbench, WEKA, which permits rapid experimentation on a given dataset using a variety of machine learning schemes, and has several facilities for interactive investigation of the data: preprocessing attributes, evaluating and comparing the results of different schemes, and designing comparative experiments to be run off-line. We discuss the partnership between agricultural scientist and machine learning researcher that our experience has shown to be vital to success. We review in some detail a particular agricultural application concerned with the culling of dairy herds.

**KEYWORDS:** additive manufacturing, machine learning, Design of Experiments, Data Generation

### 1.0 INTRODUCTION

The Waikato Environment for Knowledge Analysis (WEKA) is a New Zealand government-sponsored initiative to investigate the application of machine learning to economically important problems in the agricultural industries. The overall goals are to create a workbench for machine learning, determine the factors that contribute towards its successful application in the agricultural industries, and develop new machine learning methods and ways of assessing their effectiveness [1-5]. This project began in 1993 and is currently working towards the fulfilment of three objectives: to design and implement the workbench, to provide case studies of applications of machine learning techniques to problems in agriculture, and to develop a methodology for evaluating generalisations in terms of their entropy [6-11].

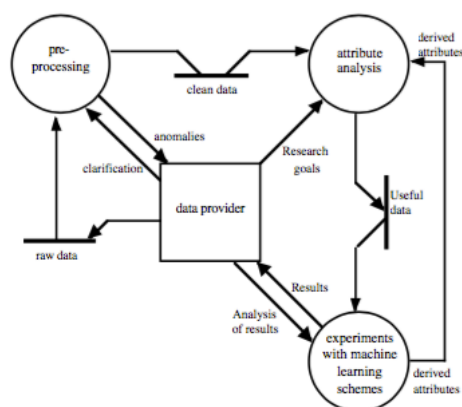


Figure 1: Process model for a machine learning application

These three objectives are by no means independent. For example, the design of the WEKA workbench has been inspired by the demands placed on it by the case studies, and has also benefited from our work on evaluating the outcomes of applying a technique to data [12-19]. Our experience throughout the development of this project is that the successful application of machine learning involves much more than merely executing a learning algorithm on some data. In this paper we present the process model that underpins our work over the past two years for the development of applications in agriculture, the software we developed around our workbench of machine learning schemes to support this model, and the outcomes and problems we have encountered in developing applications [20-26].

## 2.0 LITERATURE REVIEW

The process model we use for machine learning applications is presented as a data flow diagram in Figure 1. It is crucially dependent on two-way interaction between the provider of the data and the machine learning researcher, and enthusiasm on both sides plays a major role in ensuring that the model works. These two participants have quite different motivations for seeing the application through, and it is important that they are prepared to learn something from each other. The machine learning researcher, for example, must care more about finding something interesting in the data than building a new tool to analyse it [27-35]. This requires discipline: tool-building is something that computer scientists gladly take on, whereas finding things in data involves liaison, listening, questioning, and perhaps failure—activities that they are not so comfortable with. We receive the agricultural data that we analyse from either agricultural research institutes or private companies. In soliciting data, we find that we must first “sell” the concept of machine learning to research scientists, who tend to be wary of non-standard or non-statistical approaches. One useful technique is to stress the commonalities between statistics and machine learning, presenting them as complementary rather than promoting machine learning as a replacement for statistical analysis. A second is to point out that machine learning is useful for generating hypotheses, which can then be supported by further statistical analysis or scientific testing [36-42]. In the initial visits to an agricultural institute, we emphasise the need to establish a partnership in analysing the data. We provide expertise in applying machine learning algorithms, but the data providers must supply domain information to effectively and efficiently guide analysis. Machine learning is not a “one-shot” operation, we generally must collaborate closely with those who gathered the data. Having obtained the raw data, it must be massaged into a form suitable for processing by the automated tools. In the case of the WEKA system, the data is extracted and translated into a standard format we call ARFF, for Attribute Relation File Format. This generally involves taking the physical extract of a database and processing it through a series of steps to generate an ARFF dataset. Anomalies arise in this process and must be resolved via consultation with the data provider [43-49]. This may result in new data being generated, or in a better understanding of the values and attributes stored in the database. The data must be transformed into the single flat table required by the algorithms in WEKA. Data is commonly obtained by retrieving it from a relational database using some form of query language (e.g., SQL). This often involves taking data from a variety of sources or tables in the database and combining them into a single relation. This process, called denormalization, is the reverse of the process that was used to structure the database in the first place to impose integrity constraints, reduce update anomalies and eliminate data redundancy. The problems arising from denormalization are two-fold. First, the dataset that is extracted can be very large, in terms of both the number of columns or attributes and the number of rows or examples. The machine learning software and the computer system it is running on must both have sufficient resources to cope with the data. Second, denormalization may reintroduce functional dependencies between attributes that were the very reason for structuring the database in the first place [50-58]. If these relationships are identified as patterns by the system, they will not be considered “interesting” by the user. Most machine learning tools will introduce these two problems because they only work with a single relation, though some techniques, for example FOIL, use first-order logic and work with multiple relations. Once the database is in a single relation, each attribute must be examined to determine its data type—for example, whether it contains numeric or symbolic information. Numeric values may include measurements, such as the diameter of an apple, while symbolic values could be the day of the week or whether a cow has had a calf this year. This information may be ascertained from the original database’s data dictionary (if there is one), but there are a number of pitfalls. Databases that have been designed for custom software may contain numeric values that are actually not to be treated as numbers but rather as codes describing a particular condition [59-66]. For example, a field *production index* may have the value -1 to indicate that no measurement was recorded. In such cases, one may replace the -1 with a “missing value” token, or if that is not possible then simply remove those records. Another example is a field in the database that is of type integer but whose contents are not used arithmetically. This may arise in the case of an *identification number* field, for which certain operations—such as taking the average of the field’s values—are meaningless. Changing the field to one where the numbers are treated as nominal values will eliminate the possibility of the system creating inappropriate rules. Conversely, symbolic values may have an associated ordering—for example, the days of the week—in which case it may be better to restructure the attribute into one that allows rules to be produced that express situations such as *working\_day ≤ Tuesday* rather than *working\_day is Monday* or *working\_day is Tuesday*. Mapping the symbolic values to integers will

allow such operations, though one must be careful not to permit arithmetic or statistical operations to be applied to the integers. The user preparing the dataset needs to be aware of co-dependent and implied attributes. The former occur when one attribute contains a measurement and another indicates how accurate that measurement is. The second attribute might be a necessary qualification on the first one—for example only use the measurement if its accuracy exceeds 95%. The latter occur when the absence of a value in one attribute implies its existence in another attribute. For example, in the cow culling dataset described in Section 3, the absence of a value in the *transfer in date 3* field means check the *transfer in date 2* field to determine when the cow was transferred into the herd. Missing values in the dataset can create a variety of problems. Sometimes they merely indicate the absence of information; other times they actually convey information. In the first case all missing values should be replaced by a token recognised by the WEKA system [67-74]. In ARFF, missing values are represented by “?”. The presence of the missing value token means that the system will avoid creating rules for which the absence of a value determines the classification. On the other hand, a missing value may convey information—for example, in an attribute that contains disease information the absence of a value could signal *No disease*. Sometimes, both cases occur together in a single attribute, and it may not be possible to distinguish a missing value from a default one. Real data is surprisingly dirty, and requires sustained effort to check for consistency in measurements, locate erroneous values, and determine sources of noise. Again, this process goes more smoothly when working closely with the data providers. As with any data analysis technique, the higher the data quality, the better the model will be. If many attribute values are missing or corrupt, the schemes may not have enough data to build a complete model [75-81]. Two problems have been encountered while working with agricultural datasets. The first is that classes may overlap when different classes have very similar attribute values. Figure 2 shows the results of evaluating a model describing apple bruise size created using C4.5 on a dataset of 1440 instances. The model misclassified 404 apples in. Conversely, symbolic values may have an associated ordering—for example, the days of the week—in which case it may be better to restructure the attribute into one that allows rules to be produced that express situations such as *working\_day ≤ Tuesday* rather than *working\_day is Monday* or *working\_day is Tuesday*. Mapping the symbolic values to integers will allow such operations, though one must be careful not to permit arithmetic or statistical operations to be applied to the integers. The user preparing the dataset needs to be aware of co-dependent and implied attributes. The former occur when one attribute contains a measurement and another indicates how accurate that measurement is. The second attribute might be a necessary qualification on the first one—for example only use the measurement if its accuracy exceeds 95%. The latter occur when the absence of a value in one attribute implies its existence in another attribute. For example, in the cow culling dataset described in Section 3, the absence of a value in the *transfer in date 3* field means check the *transfer in date 2* field to determine when the cow was transferred into the herd. Missing values in the dataset can create a variety of problems. Sometimes they merely indicate the absence of information; other times they actually convey information. In the first case all missing values should be replaced by a token recognised by the WEKA system. In ARFF, missing values are represented by “?”. The presence of the missing value token means that the system will avoid creating rules for which the absence of a value determines the classification. On the other hand, a missing value may convey information—for example, in an attribute that contains disease information the absence of a value could signal *No disease*. Sometimes, both cases occur together in a single attribute, and it may not be possible to distinguish a missing value from a default one. Real data is surprisingly dirty, and requires sustained effort to check for consistency in measurements, locate erroneous values, and determine sources of noise. Again, this process goes more smoothly when working closely with the data providers [82-87]. As with any data analysis technique, the higher the data quality, the better the model will be. If many attribute values are missing or corrupt, the schemes may not have enough data to build a complete model. Two problems have been encountered while working with agricultural datasets. The first is that classes may overlap when different classes have very similar attribute values. Figure 2 shows the results of evaluating a model describing apple bruise size created using C4.5 on a dataset of 1440 instances. The model misclassified 404 apples in the dataset, an error rate of 28.1%. The apples with tiny bruises seem to form a well-defined class, but the distinction between apples with medium bruises and those with larger bruises is not clear from the attributes in the dataset. A second problem occurs when one of several classes is much larger than the others. Known as the “small disjunct” problem, this may cause machine learning schemes to describe the data as a single class and treat the instances in the smaller classes as anomalies or errors. For example, in one dataset describing cows in heat, 98% of the examples are not in heat and the remaining 2% are not clearly distinguishable using the attributes in the dataset. Thus the schemes come up with the general rule that a cow is not in heat, which is correct 98% of the time! Work is continuing in this

area. One approach is to reduce the number of instances of the larger class in the training set while maintaining the numbers of instances in the smaller classes. Another is to construct a hybrid scheme by augmenting C4.5 with an instance-based learner (Ting 1994).

Classified as				Actual
tiny	small	medium	large	Class
302	11	2		tiny
10	41	59	1	small
	4	663	26	medium
		291	30	large

Figure 2: Confusion matrix for apple bruising

At this point, we have cleaned up rows (data instances) and now need to determine the columns that are most likely to provide information and the one to be used for classification. The classification relates to the overall research goals of the data provider. These goals may have to be “extracted” from them from a knowledge of what can be achieved by using the learning schemes [1-13]. This extraction process is less painful if the data providers have been well briefed on the technology when the partnership was created. Finding useful attributes is a problem of tractability: real datasets have thousands, perhaps millions, of records, each of which may have hundreds of fields. Some schemes cannot handle the huge numbers of hypotheses that must be calculated in analysing such a dataset; others cannot handle them in a timely fashion. Pre-selection of potentially relevant attributes is critical, as omitting a key feature or including irrelevant ones can both lead to poor classification accuracy. We run the data through one or more of the following schemes to select the most relevant features: 1R, the single-attribute classifier developed by Holte (Holte 1993). While Holte uses 1R as a stand-alone learning scheme, we view it as a feature selector. Applying 1R iteratively with each of the attributes in the raw data allows us to rank attributes by their classificatory power [14-21]. C4.5, a learning scheme that has proven in practice to be quite effective in winnowing out irrelevant attributes (Quinlan 1992). We apply C4.5 directly to the initial data table, and eliminate attributes that do not play a significant role in the resulting decision tree. As will be discussed in Section 4, we also gain quick insight into possible problems with the format of key attributes. FSS, the feature selection algorithm built into the MLC++ system (Fu 1968; Kohavi *et al* 1994). FSS is a forward sequential selection algorithm which iteratively adds attributes and tests their effectiveness/relevance with an induction algorithm [22-29]. It produces a list of what the induction algorithm considers to be the most relevant attributes. Runic (Smith and Holmes 1995), an algorithm for subset selection that examines rough numeric dependencies in the data. An attribute is considered potentially relevant if a particular subrange of its values can be used to predict the value of another feature more accurately than by pure chance. The degree of predictivity can be used to rank attributes, and the best predictors chosen are for analysis by machine learning algorithms. The result of these processing steps is a table of (probably) useful attributes and consistent instances. This data is placed in an experimental loop across several schemes. In the absence of a comprehensive set of empirical tests to determine a single best learning algorithm to apply to a given dataset, we find it most effective to apply more than one scheme [30-36]. Different algorithms give different insights into the dataset, and suggest further manipulations to perform on it. We currently use the most common metric— classification accuracy—to determine the overall effectiveness of an experiment, and are developing entropy-based complexity measures for rule sets. In dealing with our data providers, we must also consider a metric that is not easily quantified: the “interestingness” of the derived rules. As we take our results back to the domain experts, we are constantly reminded that it is less than impressive to proudly present them with common-sense facts from their field that we have calculated at great effort! A raw decision tree or set of rules needs to be processed by a human, to tease out the useful or novel information from the already-known. One outcome of presenting results is the establishment of “derived attributes”—attributes that are used in practice but are not directly represented in the data [37-46]. For example, *year of birth* may not be as useful as *current age*, if the dataset includes several years of values; the production index of a cow relative to the herd average may well be more useful than the absolute production index, and so on. This step usually involves creating new columns from old ones. The most common operations include quantising a numeric attribute into intervals or classes; calculating a new attribute using a formula; and creating a new class attribute based on a series of logical conditions [47-52].

### 3. RESEARCH METHODOLOGY

The WEKA workbench has been designed to facilitate the processing of data from A R F F format to its final representation as a set of rules or decision tree. A variety of state-of-the-art, robust, and efficient machine learning schemes are provided for experimentation. The workbench is not, however, a multi-paradigm learner; that is, it does not combine machine learning techniques to produce new hybrid schemes. Instead, it concentrates on simplifying access to standard schemes so that their performance can be evaluated and compared. The design of the workbench is something of a compromise between the needs of machine learning researchers and domain experts with end-user computing experience. We have tried to avoid alienating one or another of these groups of users. This usually means that we develop software that can be used either at the command line level, which is the preferred interface for expert researchers, or at the graphical user interface level, which is the preferred mode of access for domain experts and researchers new to the technology. Papers describing the workbench have already appeared in the literature [1-9]. In this paper we concentrate on the supporting software that has been developed as a result of experience with real-world data. Existing data must be converted to our standard format, ARFF, before it can be imported into WEKA. We developed a collection of stand-alone programs that can be used to process standard formats such as INGRES files, but special-purpose code must be developed for obscure (or unique!) formats. Once the data is loaded into WEKA, it can be viewed with simple visualisation tools such as histograms and box plots. We also provide a spreadsheet for viewing data. Data can be modified using the attribute editor, which provides facilities for creating derived attributes and generating new ARFF files (Figure 3). Formulae used to define derived attributes can be arbitrarily complex, and may apply logical and numeric operators to one or more existing attributes. In generating new ARFF files, the user can select subsets of both rows and columns from the original table. The attribute editor also allows the user to append comments recording the rationale behind new attributes. Of the techniques for feature selection discussed in Section 2.2, only 1R and C4.5 are currently part of the workbench proper; we run FSS and Runic as stand-alone programs. We are currently considering how best to add these latter two algorithms to the workbench [10-19].

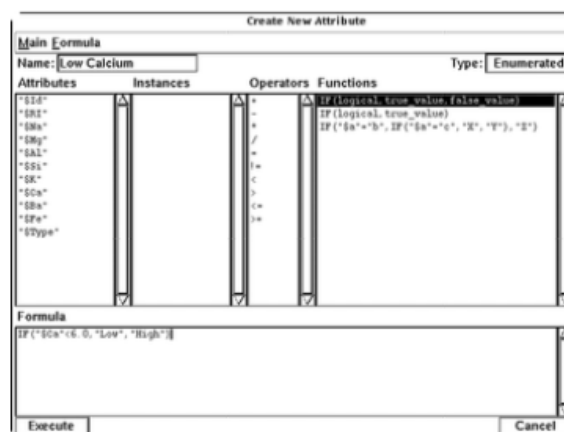


Figure 3. Attribute editor for new attribute creation

The experiment editor (Figure 4) is a simple graphical user interface tool for organising and running several machine learning schemes over several sets of data. It is a more general implementation of Holte's experimental paradigm [20-27]. A user specifies the ratio of the size of test and training sets and the number of runs required (for example, Holte uses a 1/3:2/3 split of data and 25 runs). Datasets are chosen by the user, who is asked to specify the attribute to be used for classification. Schemes are chosen from those provided by the workbench. Once these items have been selected, the system randomly generates the required number of test and training files, and runs each of the selected schemes on this data. Results are collated in a text file and processed to provide summary statistics from the PREval evaluator (see below). All output from a machine learning scheme is passed back to WEKA in the form of text, and can be displayed in a scrollable viewer. Text can be copied to other applications, printed, or saved to a file. A novel feature of the workbench is that it supplies a facility for

“external” evaluation that can perform tests on the output from any machine learning scheme in a uniform way. If the user selects external evaluation, the output, as well as being displayed as text, is converted into an internal rule format and evaluated. The rule format is PROLOG-based, and rules can be executed using an evaluator called PREval. The precise details of output translation varies for the individual schemes. For FOIL, which produces its output as rules already, it is a simple matter to convert the rule format to PREval. INDUCT, which produces rules in a special “ripple-down” structure, provides an option to produce its output directly in the PREval format [28-34]. For C4.5, both decision trees and rules are converted automatically into PREval rules.



Figure 4. WEKA experiment editor

PREval takes a set of rules and an ARFF file, and evaluates how well the rules cover the classifications. It provides figures for classification accuracy, including the percentage correctly classified, incorrectly classified, classified by multiple rules, and not classified at all, as well as confusion matrices to show the distribution of misclassifications, and statistics on how well each rule does [35-42]. PREval also incorporates an entropy measure for calculating both the complexity of rule sets, and the complexity of datasets with respect to a rule set. PREval takes a set of rules and an ARFF file, and evaluates how well the rules cover the classifications. It provides figures for classification accuracy, including the percentage correctly classified, incorrectly classified, classified by multiple rules, and not classified at all, as well as confusion matrices to show the distribution of misclassifications, and statistics on how well each rule does. PREval also incorporates an entropy measure for calculating both the complexity of rule sets, and the complexity of datasets with respect to a rule set [43-52].

## 4. RESULT

In this section we examine a project in which WEKA has been used to find rules describing culling decisions in dairy herds. We relate our development of the application to the process model and the tools we described in sections 2 and 3. The “cow culling” database was provided by the Livestock Improvement Corporation in Hamilton, New Zealand. This institution operates a relational database system to track genetic history and production records of twelve million dairy cows and sires, of which three million are currently alive. Production data are recorded for each cow from four to twelve times per year, and additional information is recorded as events occur. Farmers receive information from the Livestock Improvement Corporation in the form of reports from which comparisons within the herd can be made. Two types of information that are produced are the *production* and *breeding indexes* (PI and BI respectively), which both indicate the merit of the animal. The former reflects the milk it produces with respect to measures such as fat, protein and volume, indicating its worth as a production animal. The latter reflects the likely merit of a cow’s progeny, indicating its worth as a breeding

animal. In a well-managed herd, the average value of these indexes will increase every year, as superior animals enter the herd and low- performance ones are removed.

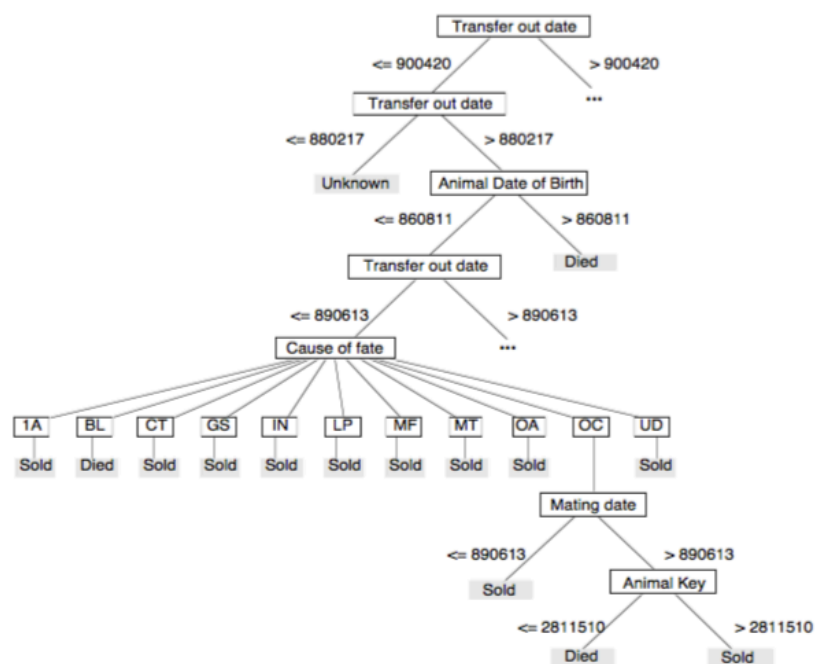


Figure 5. Decision tree induced from raw herd data

One major decision that farmers must make each year is whether to retain a cow in the herd or remove it, usually to an abattoir. About 20% of the cows in a typical New Zealand dairy herd are culled each year, usually near the end of the milking season as feed reserves run short. The cows' breeding and production indexes influence this decision, particularly when compared with the other animals in the herd. Other factors that may influence the decision are:

- age: a cow is nearing the end of its productive life at 8–10 years,
- health problems,
- history of difficult calving,
- undesirable temperament traits (kicking, jumping fences),
- not being in calf for the following season.

The Livestock Improvement Corporation hoped that machine learning tools would provide insight into the rules that particular farmers actually use to make their culling decisions, enabling better information to be provided to farmers in the future. They provided data from ten herds over six years, representing 19 000 records each containing 705 attributes. The initial dataset we studied was extremely noisy, and indicated the need to include simple data visualisation techniques such as histograms in our workbench. A number of attributes contained a predominance of missing values; these could be quickly identified through visualisation, and were eliminated. These techniques are also helpful in determining viable clusters when discretizing real values. When the data was originally denormalized into a single table, the link and join relations duplicated some attributes (under different names). These were identified and deleted. The principal tools used to select potentially relevant attributes were C4.5 (Quinlan 1992) and FOIL. The initial flat table was run through C4.5 on the workbench. Cows were classified on their *fate code* attribute, which can take the values *sold*, *dead*, *lost* and *unknown*. The resulting tree, shown in Figure 5, proved disappointing. At its root is the *transfer out date* attribute. This implies that the culling decision for a particular cow is based mainly on the date on



which it is culled, rather than on any attributes of the cow. Next, the *date of birth* is used, but as the culling decisions take place in different years, an absolute date is not meaningful. A cow's age would be useful, but is not explicitly present in the dataset. The *cause of fate* attribute is strongly associated with the *fate code*; it contains a coded explanation of the reason for culling. This attribute is assigned a value *after* the culling decision is made, so it is not available to the farmer who makes the culling decision. Furthermore, we would like to be able to predict this attribute—in particular the *low production* (LP) value—rather than include it in the tree as a decision indicator. Its presence artificially elevated the classification accuracy, predicting the culling decision correctly 95% of the time on test data. *Mating date* is another absolute date attribute, and animal key is simply a 7-digit identifier. In discussions with staff from the Livestock Improvement Corporation, it was suggested that the culling decision may not be based on a cow's absolute performance, but on its performance relative to the rest of the herd. To test this, attributes representing the difference in production from the average over the cow's herd were added to the database. In order not to bias the learning process, the original attributes were retained, and the new attributes were not distinguished in any way—it was left to the machine learning schemes to decide if they were more helpful for classification than the original ones. This process involved close cooperation with Livestock Improvement Corporation. Discussions often ended up proposing more derived attributes, and clarifying the meaning of particular attributes. Staff were also able to evaluate the plausibility of rules, which was helpful in the early stages when the recreation of existing knowledge was a useful indication of the correctness of our approach. It was necessary to create about forty new attributes, including a status code showing whether a cow is retained or culled and a variety of production and breeding indices relative to the herd in a specific year. This experience was one of the primary motivating factors for the construction of the attribute editor described in Section 3. After adding derived attributes, C4.5 produced the tree in Figure 6. The *fate code*, *cause of fate* and *transfer out date* have been combined into a *status code* which takes the values *culled* or *retained*. For each year, the records for cows that have previously been culled or have not yet been born are removed. Cows that were alive and were not transferred in that year are marked as *retained*, otherwise they are marked *culled*. If, however, a cow died of disease or some other factor outside the farmer's control, the record is removed. After all, the aim of this exercise is to discover the farmer's culling rules rather than the incidence of disease and injury. The tree in Figure 6 is much more compact than the one in Figure 5. It was produced with 30% of the instances, and correctly classifies 95% of the remaining ones. The unconditional retention of cows two years or younger is due to the fact that they have not yet begun lactation, and so there is no indication of their productive potential. The next decision is based on the cow's worth as a breeding animal, which is calculated from the earnings of its offspring. The volume of milk that the cow produces is used for the final decision. This tree's decisions are plausible from a farming perspective, and its compactness indicates that it is a good explanation of the culling process. It is interesting to note that it consists entirely of derived attributes, further emphasising the importance of this step in the process.

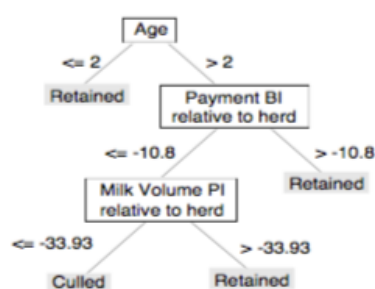


Figure 6. Cow culling decision tree

## 5. CONCLUSION

The discovery process is an iterative one. After taking a dataset and analysing it, the results are used to augment the dataset with new attributes or to adjust a scheme's parameters. Then the process is repeated. The WEKA workbench supports this process by providing the user with uniform facilities for



manipulating datasets, configuring learning schemes and handling output. But while a good set of tools is important, forming a partnership with data providers is crucial. Domain knowledge is necessary to analyse data effectively, and our discussions with agricultural experts direct data processing, experimentation, and interpretation of results. Our experience with New Zealand agricultural datasets has suggested several other improvements for the workbench. We tend to run a large number of experiments on a single dataset: creating new attributes, selecting subsets of rows and columns for processing, running the derived datasets across several machine learning schemes, testing different parameters on a single scheme, etc. Since it quickly becomes difficult to remember which combination of attributes produced what results on which algorithm, we are considering incorporating a scheme for organizing experimental results into the workbench. Similar facilities have recently been designed for statistics packages, and it appears that these representation techniques are also applicable to machine learning analysis. A longer-term goal would be to provide a method for codifying *a priori* domain knowledge and incorporating it into the machine learning model in a principled fashion. As discussed in Section 2, we use prior knowledge of the domain to create new (derived) attributes, to eliminate attributes that are unlikely to be useful, and to guide the experimentation process. Most of this information is not recorded, or if recorded is not easily associated with the relevant dataset. We have implemented some simple solutions such as adding a comment field to the attribute editor so that we can keep track of what derived attributes represent. A much more difficult problem is to provide suggestions for creating derived attributes, or for effectively using domain information to incorporate a bias into the learning process.

## REFERENCES

- [1] Hayami, Yujiro, and Vernon W. Ruttan. *Agricultural development: an international perspective*. Baltimore, Md/London: The Johns Hopkins Press, 1971.
- [2] Wang, Chengcheng, et al. "Machine learning in additive manufacturing: State-of-the-art and perspectives." *Additive Manufacturing* 36 (2020): 101538.
- [3] Zhu, Zuowei, et al. "Machine learning in tolerancing for additive manufacturing." *CIRP Annals* 67.1 (2018): 157-160.
- [4] Mellor, John Williams. "The economics of agricultural development." *The economics of agricultural development*. (1966).
- [5] Shafaie, Majid, Maziar Khademi, Mohsen Sarparast, and Hongyan Zhang. "Modified GTN parameters calibration in additive manufacturing of Ti-6Al-4 V alloy: a hybrid ANN-PSO approach." *The International Journal of Advanced Manufacturing Technology* 123, no. 11 (2022): 4385-4398.
- [6] Meng, Lingbin, et al. "Machine learning in additive manufacturing: a review." *Jom* 72.6 (2020): 2363-2377.
- [7] Staatz, John M., and Carl K. Eicher. "Agricultural development ideas in historical perspective." *International agricultural development* 3 (1998).
- [8] Sarparast, Mohsen, Majid Ghoreishi, Tohid Jahangirpoor, and Vahid Tahmasbi. "Experimental and finite element investigation of high-speed bone drilling: evaluation of force and temperature." *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 42, no. 6 (2020): 1-9.
- [9] Li, Zhixiong, et al. "Prediction of surface roughness in extrusion-based additive manufacturing with machine learning." *Robotics and Computer-Integrated Manufacturing* 57 (2019): 488-495.
- [10] Jin, Zeqing, et al. "Machine learning for advanced additive manufacturing." *Matter* 3.5 (2020): 1541-1556.
- [11] Jiang, Jingchao, et al. "Machine learning integrated design for additive manufacturing." *Journal of Intelligent Manufacturing* (2020): 1-14.
- [12] Nakasone, Eduardo, Maximo Torero, and Bart Minten. "The power of information: The ICT revolution in agricultural development." *Annu. Rev. Resour. Econ.* 6.1 (2014): 533-550.
- [13] Sarparast, Mohsen, Majid Ghoreishi, Tohid Jahangirpoor, and Vahid Tahmasbi. "Modelling and optimisation of temperature and force behaviour in high-speed bone drilling." *Biotechnology & Biotechnological Equipment* 33, no. 1 (2019): 1616-1625.
- [14] Baumann, Felix W., et al. "Trends of machine learning in additive manufacturing." *International Journal of Rapid Manufacturing* 7.4 (2018): 310-336.
- [15] Gu, Grace X., et al. "Bioinspired hierarchical composite design using machine learning: simulation, additive manufacturing, and experiment." *Materials Horizons* 5.5 (2018): 939-945.
- [16] Pinstrip-Andersen, Per, and Satoru Shimokawa. "Rural infrastructure and agricultural development." *World Bank* (2006).
- [17] Bahrami, Javad, Mohammad Ebrahimabadi, Sofiane Takarabt, Jean-luc Danger, Sylvain Guille, and Naghmeh Karimi. "On the Practicality of Relying on Simulations in Different Abstraction Levels for Pre-Silicon Side-Channel Analysis."
- [18] Qi, Xinbo, et al. "Applying neural-network-based machine learning to additive manufacturing: current applications, challenges, and future perspectives." *Engineering* 5.4 (2019): 721-729.

- [19] Johnson, N. S., et al. "Invited review: Machine learning for materials developments in metals additive manufacturing." *Additive Manufacturing* 36 (2020): 101641.
- [20]
- [21] Sadi, Mehdi, Yi He, Yanjing Li, Mahabubul Alam, Satwik Kundu, Swaroop Ghosh, Javad Bahrami, and Naghmeh Karimi. "Special Session: On the Reliability of Conventional and Quantum Neural Network Hardware." In 2022 IEEE 40th VLSI Test Symposium (VTS), pp. 1-12. IEEE, 2022.
- [22] Qin, Jian, et al. "Research and application of machine learning for additive manufacturing." *Additive Manufacturing* (2022): 102691.
- [23] Caggiano, Alessandra, et al. "Machine learning-based image processing for on-line defect recognition in additive manufacturing." *CIRP annals* 68.1 (2019): 451-454.
- [24] Norton, George W., and Jeffrey Alwang. *Introduction to economics of agricultural development*. McGraw-Hill, Inc., 1993.
- [25] Bahrami, Javad, Mohammad Ebrahimabadi, Jean-Luc Danger, Sylvain Guilley, and Naghmeh Karimi. "Leakage power analysis in different S-box masking protection schemes." In 2022 Design, Automation & Test in Europe Conference & Exhibition (DATE), pp. 1263-1268. IEEE, 2022.
- [26] Yao, Xiling, Seung Ki Moon, and Guijun Bi. "A hybrid machine learning approach for additive manufacturing design feature recommendation." *Rapid Prototyping Journal* (2017).
- [27] Zhao, Jingzhu, et al. "Opportunities and challenges of sustainable agricultural development in China." *Philosophical Transactions of the Royal Society B: Biological Sciences* 363.1492 (2008): 893-904.
- [28] Bahrami, Javad, Viet B. Dang, Abubakr Abdulgadir, Khaled N. Khasawneh, Jens-Peter Kaps, and Kris Gaj. "Lightweight implementation of the lowmc block cipher protected against side-channel attacks." In *Proceedings of the 4th ACM Workshop on Attacks and Solutions in Hardware Security*, pp. 45-56. 2020.
- [29] Sing, S. L., et al. "Perspectives of using machine learning in laser powder bed fusion for metal additive manufacturing." *Virtual and Physical Prototyping* 16.3 (2021): 372-386.
- [30] Ahmadinejad, Farzad, Javad Bahrami, Mohammad Bagher Menhaj, and Saeed Shiry Ghidary. "Autonomous Flight of Quadcopters in the Presence of Ground Effect." In 2018 4th Iranian Conference on Signal Processing and Intelligent Systems (ICSPIS), pp. 217-223. IEEE, 2018.
- [31] Ko, Hyunwoong, et al. "Machine learning and knowledge graph based design rule construction for additive manufacturing." *Additive Manufacturing* 37 (2021): 101620.
- [32] Hadiana, Hengameh, Amir Mohammad Golmohammadib, Hasan Hosseini Nasabc, and Negar Jahanbakhsh Javidd. "Time Parameter Estimation Using Statistical Distribution of Weibull to Improve Reliability." (2017).
- [33] Jiang, Jingchao, et al. "Achieving better connections between deposited lines in additive manufacturing via machine learning." *Math. Biosci. Eng* 17.4 (2020): 3382-3394.
- [34] Zavareh, Bozorgasl, Hossein Foroozan, Meysam Gheisarnejad, and Mohammad-Hassan Khooban. "New trends on digital twin-based blockchain technology in zero-emission ship applications." *Naval Engineers Journal* 133, no. 3 (2021): 115-135.
- [35] Li, Rui, Mingzhou Jin, and Vincent C. Paquit. "Geometrical defect detection for additive manufacturing with machine learning models." *Materials & Design* 206 (2021): 109726.
- [36] Baturynska, Ivanna, Oleksandr Semeniuta, and Kristian Martinsen. "Optimization of process parameters for powder bed fusion additive manufacturing by combination of machine learning and finite element method: A conceptual framework." *Procedia Cirp* 67 (2018): 227-232.
- [37] Bozorgasl, Zavareh, and Mohammad J. Dehghani. "2-D DOA estimation in wireless location system via sparse representation." In 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), pp. 86-89. IEEE, 2014.
- [38] Gobert, Christian, et al. "Application of supervised machine learning for defect detection during metallic powder bed fusion additive manufacturing using high resolution imaging." *Additive Manufacturing* 21 (2018): 517-528.
- [39] Amini, Mahyar, Koosha Sharifani, and Ali Rahmani. "Machine Learning Model Towards Evaluating Data gathering methods in Manufacturing and Mechanical Engineering." *International Journal of Applied Science and Engineering Research* 15.4 (2023): 349-362.
- [40] Paul, Arindam, et al. "A real-time iterative machine learning approach for temperature profile prediction in additive manufacturing processes." 2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA). IEEE, 2019.
- [41] Amini, Mahyar, and Aryati Bakri. "Cloud computing adoption by SMEs in the Malaysia: A multi-perspective framework based on DOI theory and TOE framework." *Journal of Information Technology & Information Systems Research (JITISR)* 9.2 (2015): 121-135.
- [42] Mahmoud, Dalia, et al. "Applications of Machine Learning in Process Monitoring and Controls of L-PBF Additive Manufacturing: A Review." *Applied Sciences* 11.24 (2021): 11910.
- [43] Amini, Mahyar. "The factors that influence on adoption of cloud computing for small and medium enterprises." (2014).
- [44] Khanzadeh, Mojtaba, et al. "Quantifying geometric accuracy with unsupervised machine learning: Using self-organizing map on fused filament fabrication additive manufacturing parts." *Journal of Manufacturing Science and Engineering* 140.3 (2018).
- [45] Amini, Mahyar, et al. "MAHAMGOSTAR.COM AS A CASE STUDY FOR ADOPTION OF LARAVEL

- FRAMEWORK AS THE BEST PROGRAMMING TOOLS FOR PHP BASED WEB DEVELOPMENT FOR SMALL AND MEDIUM ENTERPRISES." *Journal of Innovation & Knowledge*, ISSN (2021): 100-110.
- [46] Liu, Rui, Sen Liu, and Xiaoli Zhang. "A physics-informed machine learning model for porosity analysis in laser powder bed fusion additive manufacturing." *The International Journal of Advanced Manufacturing Technology* 113.7 (2021): 1943-1958.
- [47] Amini, Mahyar, et al. "Development of an instrument for assessing the impact of environmental context on adoption of cloud computing for small and medium enterprises." *Australian Journal of Basic and Applied Sciences (AJBAS)* 8.10 (2014): 129-135.
- [48] Nojeem, Lolade, et al. "Machine Learning assisted response circles for additive manufacturing concerning metal dust bed fusion." *International Journal of Management System and Applied Science* 17.11 (2022): 491-501.
- [49] Hu, Dinghuan, et al. "The emergence of supermarkets with Chinese characteristics: challenges and opportunities for China's agricultural development." *Development policy review* 22.5 (2004): 557-586.
- [50] Golmohammadi, Amir-Mohammad, Negar Jahanbakhsh Javid, Lily Poursoltan, and Hamid Esmaeeli. "Modeling and analyzing one vendor-multiple retailers VMI SC using Stackelberg game theory." *Industrial Engineering and Management Systems* 15, no. 4 (2016): 385-395.
- [51] Motalo, Kubura, et al. "evaluating artificial intelligence effects on additive manufacturing by machine learning procedure." *Journal of Basis Applied Science and Management System* 13.6 (2023): 3196-3205.
- [52] Fu, Yanzhou, et al. "Machine learning algorithms for defect detection in metal laser-based additive manufacturing: a review." *Journal of Manufacturing Processes* 75 (2022): 693-710.
- [53] Amini, Mahyar, et al. "The role of top manager behaviours on adoption of cloud computing for small and medium enterprises." *Australian Journal of Basic and Applied Sciences (AJBAS)* 8.1 (2014): 490-498.
- [54] Stathatos, Emmanuel, and George-Christopher Vosniakos. "Real-time simulation for long paths in laser-based additive manufacturing: a machine learning approach." *The International Journal of Advanced Manufacturing Technology* 104.5 (2019): 1967-1984.
- [55] Amini, Mahyar, and Nazli Sadat Safavi. "Critical success factors for ERP implementation." *International Journal of Information Technology & Information Systems* 5.15 (2013): 1-23.
- [56] Petrich, Jan, et al. "Multi-modal sensor fusion with machine learning for data-driven process monitoring for additive manufacturing." *Additive Manufacturing* 48 (2021): 102364.
- [57] Amini, Mahyar, et al. "Agricultural development in IRAN base on cloud computing theory." *International Journal of Engineering Research & Technology (IJERT)* 2.6 (2013): 796-801.
- [58] Snow, Zackary, et al. "Toward in-situ flaw detection in laser powder bed fusion additive manufacturing through layerwise imagery and machine learning." *Journal of Manufacturing Systems* 59 (2021): 12-26.
- [59] Amini, Mahyar, et al. "Types of cloud computing (public and private) that transform the organization more effectively." *International Journal of Engineering Research & Technology (IJERT)* 2.5 (2013): 1263-1269.
- [60] Akhil, V., et al. "Image data-based surface texture characterization and prediction using machine learning approaches for additive manufacturing." *Journal of Computing and Information Science in Engineering* 20.2 (2020): 021010.
- [61] Amini, Mahyar, and Nazli Sadat Safavi. "Cloud Computing Transform the Way of IT Delivers Services to the Organizations." *International Journal of Innovation & Management Science Research* 1.61 (2013): 1-5.
- [62] Douard, Améline, et al. "An example of machine learning applied in additive manufacturing." *2018 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*. IEEE, 2018.
- [63] Amini, Mahyar, and Nazli Sadat Safavi. "A Dynamic SLA Aware Heuristic Solution For IaaS Cloud Placement Problem Without Migration." *International Journal of Computer Science and Information Technologies* 6.11 (2014): 25-30.
- [64] Ding, Donghong, et al. "The first step towards intelligent wire arc additive manufacturing: An automatic bead modelling system using machine learning through industrial information integration." *Journal of Industrial Information Integration* 23 (2021): 100218.
- [65] Amini, Mahyar, and Nazli Sadat Safavi. "A Dynamic SLA Aware Solution For IaaS Cloud Placement Problem Using Simulated Annealing." *International Journal of Computer Science and Information Technologies* 6.11 (2014): 52-57.
- [66] Zhang, Wentai, et al. "Machine learning enabled powder spreading process map for metal additive manufacturing (AM)." *2017 International Solid Freeform Fabrication Symposium*. University of Texas at Austin, 2017.
- [67] Sharifani, Koosha and Amini, Mahyar and Akbari, Yaser and Aghajanzadeh Godarzi, Javad. "Operating Machine Learning across Natural Language Processing Techniques for Improvement of Fabricated News Model." *International Journal of Science and Information System Research* 12.9 (2022): 20-44.
- [68] Du, Y., T. Mukherjee, and T. DebRoy. "Physics-informed machine learning and mechanistic modeling of additive manufacturing to reduce defects." *Applied Materials Today* 24 (2021): 101123.
- [69] Sadat Safavi, Nazli, et al. "An effective model for evaluating organizational risk and cost in ERP implementation by SME." *IOSR Journal of Business and Management (IOSR-JBM)* 10.6 (2013): 70-75.
- [70] Chen, Lequn, et al. "Rapid surface defect identification for additive manufacturing with in-situ point cloud processing and machine learning." *Virtual and Physical Prototyping* 16.1 (2021): 50-67.
- [71] Sadat Safavi, Nazli, Nor Hidayati Zakaria, and Mahyar Amini. "The risk analysis of system selection and

- business process re-engineering towards the success of enterprise resource planning project for small and medium enterprise." *World Applied Sciences Journal (WASJ)* 31.9 (2014): 1669-1676.
- [72] Yaseer, Ahmed, and Heping Chen. "Machine learning based layer roughness modeling in robotic additive manufacturing." *Journal of Manufacturing Processes* 70 (2021): 543-552.
- [73] Sadat Safavi, Nazli, Mahyar Amini, and Seyyed AmirAli Javadinia. "The determinant of adoption of enterprise resource planning for small and medium enterprises in Iran." *International Journal of Advanced Research in IT and Engineering (IJARIE)* 3.1 (2014): 1-8.
- [74] Chen, Lequn, et al. "Rapid surface defect identification for additive manufacturing with in-situ point cloud processing and machine learning." *Virtual and Physical Prototyping* 16.1 (2021): 50-67.
- [75] Safavi, Nazli Sadat, et al. "An effective model for evaluating organizational risk and cost in ERP implementation by SME." *IOSR Journal of Business and Management (IOSR-JBM)* 10.6 (2013): 61-66.
- [76] Guo, Shenghan, et al. "Machine learning for metal additive manufacturing: Towards a physics-informed data-driven paradigm." *Journal of Manufacturing Systems* 62 (2022): 145-163.
- [77] Khoshraftar, Alireza, et al. "Improving The CRM System In Healthcare Organization." *International Journal of Computer Engineering & Sciences (IJCES)* 1.2 (2011): 28-35.
- [78] Ren, K., et al. "Thermal field prediction for laser scanning paths in laser aided additive manufacturing by physics-based machine learning." *Computer Methods in Applied Mechanics and Engineering* 362 (2020): 112734.
- [79] Abdollahzadegan, A., Che Hussin, A. R., Moshfegh Gohary, M., & Amini, M. (2013). The organizational critical success factors for adopting cloud computing in SMEs. *Journal of Information Systems Research and Innovation (JISRI)*, 4(1), 67-74.
- [80] Zhu, Qiming, Zeliang Liu, and Jinhui Yan. "Machine learning for metal additive manufacturing: predicting temperature and melt pool fluid dynamics using physics-informed neural networks." *Computational Mechanics* 67.2 (2021): 619-635.
- [81] Sullivan, Emily. "Understanding from machine learning models." *The British Journal for the Philosophy of Science* (2022).
- [82] Newiduum, Ladson, et al. "Machine learning model improvement based on additive manufacturing process for melt pool warmth predicting." *International Journal of Applied Science and Engineering Research* 15.6 (2023): 240-250.
- [83] Jackson, Keyipi, et al. "Increasing substantial productivity in additive manufacturing procedures by expending machine learning method." *International Journal of Mechanical Engineering and Science Study* 10.8 (2023): 363-372.
- [84] Opuiyo, Atora, et al. "Evaluating additive manufacturing procedure which was using artificial intelligence." *International Journal of Science and Information System* 13.5 (2023): 107-116.
- [85] Amini, Mahyar, and Ali Rahmani. "Machine learning process evaluating damage classification of composites." *Journal* 9.12 (2023): 240-250.
- [86] Yun, Chidi, et al. "Machine learning attitude towards temperature report forecast in additive manufacturing procedures." *International Journal of Basis Applied Science and Study* 9.12 (2023): 287-297.
- [87] Shun, Miki, et al. "Incorporated design aimed at additive manufacturing through machine learning." *International Journal of Information System and Innovation Adoption* 16.10 (2023): 823-831.