PREDICTING THE AMOUNT OF WASTAGE OF FINISHED GOODS IN TEXTILE DYEING FACTORIES

Özgür Altunsöğüt[1], Erdem Uçar[1] and Yılmaz Kılıçaslan[2]
[1]*Department of Computer Engineering, Faculty of Engineering, Trakya University, 22180 Edirne, Turkey*
[2]*Department of Computer Engineering, Faculty of Engineering, Adnan Menderes University, 09100 Aydın, Turkey*

Abstract

Fabrics undergo a series of processes in these factories that may cause defects. For this reason, the last process in the series is quality control. The position and size of the defects are determined in this control phase. Manufacturers opt for overproduction in order to compensate for the amount of wastage caused by such defects. As the production cost is high, it is important to correctly predict the overproduced amount. However, such predictions are error-prone when they are human-based. This study offers a machine-learning-based method for predicting the likelihood of wastage at certain intervals. Six different learning algorithms (J48, Naïve Bayes, KNN, SVM, Logistic Regression, and Multi-Layer Perceptron) were trained and tested on real data from a fabric dyeing factory. The experimental results show that some algorithms outperform humans in predicting the amount of wastage. The paper also offers an evaluation of the classification performances of learning models and of human predictors working in the field in order to justify the study and gain insight into the question of how to match a model to the task at hand.

Keywords: predicting, textile, wastage, machine learning, data mining

INTRODUCTION

Dyeing factories have many customer orders to produce fabrics. In order to produce finished goods these factories apply some chemical operations on raw fabrics. Each operation has its own specifications. Many factors such as temperature, speed, and pressure could cause defects in the fabrics during production. All defective parts, which are considered as wastage, have to be removed from the fabrics. Due to these defects, overproduction is inevitable. The amount of overproduction is decided by the relevant authorized person in the factories. That person makes this decision in the light of his experience. That, naturally, makes the system human-dependent. Human-dependence amounts to undue tolerance for customer orders in the case of textile production. Over-production of fabrics means loss of profit. Therefore, it is crucial to predict the amount of production as precisely as possible.

Wastage and prediction terms are using on food, energy, water, production and fuel industries. [11][12][13][14][15]. This study offers an approach to render the prediction of the amount of wastage human-independent on textile industry. The approach is based on a machine-learning method where six different learning algorithms were trained and tested on real data from a fabric dyeing factory. A series of experiments was carried out using the algorithms J48, Naïve Bayes, KNN, SVM, Logistic Regression, and Multi-Layer Perceptron. The experimental results show that some algorithms outperform humans in predicting the amount of wastage. The paper comparatively discusses the classification performances of the learning models and of the human predictors working in the field in order to justify the study and gain insight into the question of how to match a model to the task of predicting the wastage amount in textile production.

Apart from this introductory section, the organization of the paper is as follows:

Section 2 is devoted to the experimental setup of the work presented. In Section 3, the experiments and their evaluations are shared. The discussion culminates in Section 4 with a summary of the paper and some concluding remarks.

EXPERIMENTAL SETUP

We have compiled 105995 data samples from orders to a dyeing factory, which were collected in 2013. The orders were originally stored in various tables in an MS SQL database. We have transferred the data into a single Excel-like table using a set of SQL commands, a fragment of which is shown in Table 1.

Orders Amount	Wastage Amount	Wastage Ratio	P1	...	P73
5390	56.20	1.0	1	...	0
1678	247.90	14.8	0	...	1
3202	34.80	1.1	1	...	0
...	...	...	...	...	...

Table 1: Sample dataset

Each line of the table gives information about the production details of an order. The first and second columns indicate the amount of an order and of the resulting wastage for that order, respectively. The ratio of wastage appears in the third column. The remaining columns are for the 73 processes. These processes serve as learning features in our experiments. Each ordered roll of fabric undergoes a certain subset of the processes, which are marked as '1', making them positive features for that sample.

Like all sorts of data collected from a real domain, our data contain noise. The noise might be either machine or human-originated. For instance, a technically malfunctioning machine or a highly careless operator can generate a roll of defective fabric that amounts to more than 10% of the whole, such as sample 2 in Table 1. Such a ratio is never the result of a normal production process. For this reason, we eliminated all such samples from the dataset prior to the experiments. The dataset was reduced to 90480 as a result of this elimination task. Similarly, all other kinds of data deviating from normal measures were eliminated. In fact, our original data were highly contaminated, which was clearly evidenced by the fact that dataset was reduced to an amount of 11130.

Weka is used for testing algorithms. Weka is an open source Java based machine learning tool. Algorithms in Weka are tested by default parameters. All figures in this paper were generated by Weka.

Figure 1: *Before eliminating noisy data*

Figure 1 shows all data without eliminating noise. After eliminating noise, the results are shown in figure 2

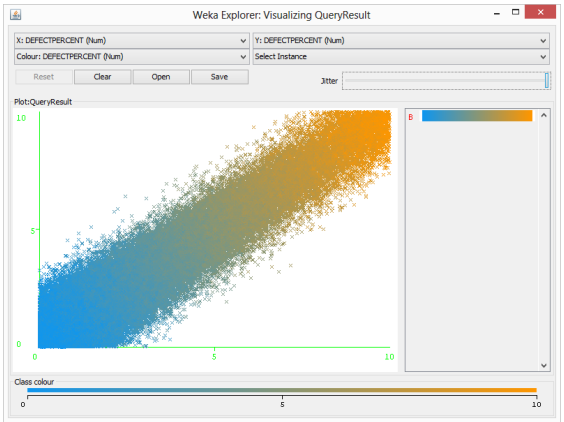

Figure 2: *After eliminating noisy data*

Defect Percent Range	Class	Instance Count
0-1	A	427
1-2	B	1.547
2-3	C	5.596
3-4	D	3.316
4-5	E	133
5-6	F	52
6-7	G	24
7-8	H	10
8-9	I	14
9-10	J	11
Total Row Count:		**11.130**

Table 2: Instance dispersion by defect percentage

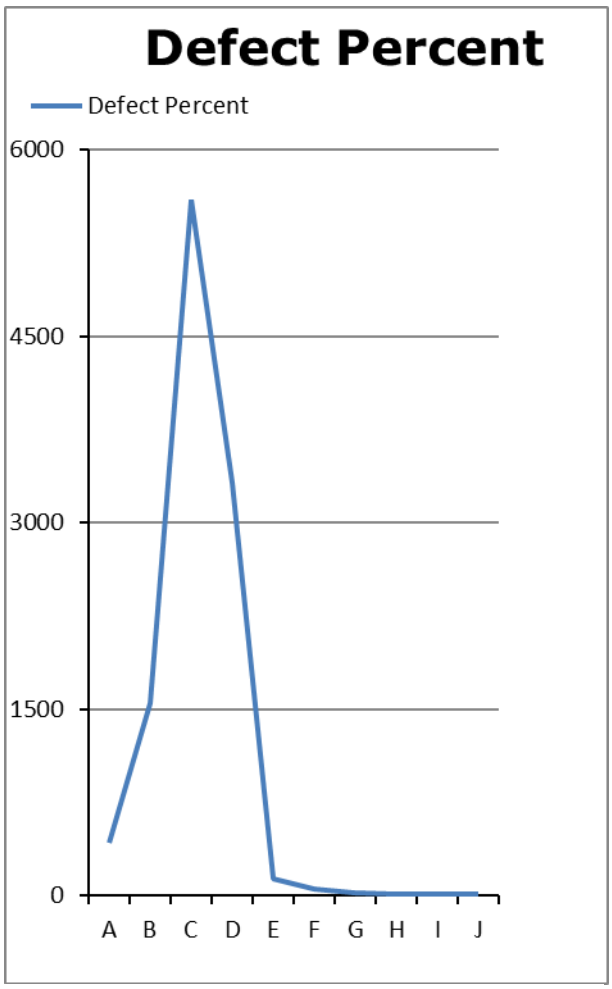

Figure 3: *Dispersion of defect percentage graph.*

Table 3: Final dataset

As seen in the figure 3 and in table 2, instances between 0% - 3% are more than 68% percent of the whole data. Underfitting the data is a problem like overfitting. Underfitting the training data by ignoring the large number of misclassified points would yield too simple models that would suffer from degraded predictive performances [10]. In order to make a balanced dataset, data that contain defects between 3% – 10% were grouped in a single class. Classification algorithms explained below are tested data in table 3.nd competitive power is supported.

Experimental Setup

Six classification algorithms were used to test the real-world dataset. A short description of the algorithms is given below.

J48

This algorithm is a special version of the C4.5 decision tree (Quinlann, 1993) algorithm in Weka. J48 uses information gain as its attribute selection.

"s" is the attributes in the data

"c" is total count of output class

Entropy(S) = $\sum_{i=1}^{c} -Pi * log2(Pi)$

Entropy(S) is general entropy for output classes.

t = Different element count for attributes.

Entropy(A) = $\sum_{i=1}^{t} -Pi * log2(Pi)$

Entropy(A) is entropy of each attribute.

Information Gain is calculated last.

GAIN(S, A) = Entropy(S) – Entropy(A)

Information gain is calculated for each attribute. The attribute that has the biggest information gain is selected and put on the top of the tree.

Naive Bayes

Naive Bayes is a linear classifier with the normal distribution assumption (Manning et al., 2008) and becomes a possibly non-linear classifier with a kernel density estimator (John and Langley, 1995). Naive Bayes classifiers are a family of simple probabilistic classifiers based on applying Bayes' theorem with strong independence assumptions between the features. The equation of the Naive Bayes theorem is given below.

$$P(A|B) = \frac{P(B|A)\,P(A)}{P(B)}$$

Defect Percent Range	Class	Instance Count
0-2	A	1.974
2-3	B	5.596
3-10	C	3.560
Total Row Count:		**11.130**

P(A|B); Probability of the A in the subset of the B

P(B|A); Probability of the B in the subset of the A

P(A) and P(B); Probability of A and B.

K's Nearest Neighbor

This algorithm is a type of instance based algorithm. The parameter k means the count of neighbor data that will be classified. Defining the k value is important for classification. Distance from the objects is calculated.

Support Vector Machine (SVM)

The SVM algorithm is a machine learning approach for classification and regression problems. So far, research progress shows that the SVM demonstrates superior

performance gains and robustness in many applications over traditional methods [1]. The main idea of support vector classification is to map the training data into a higher-dimensional space via a mapping function and construct a separating hyperplane with maximum margin there. It uses non-linear mapping to transform the original training data into a higher dimension. Within this new dimension it searches for the linear optimal separating hyperplane. The SVM finds this hyperplane using support vectors [2].

Logistic Regression

This algorithm is a type of statistical classification model. It is used for predicting class variables of the dataset. Logistic regression measures the relationship between the categorical dependent variable and one or more independent variables, which are usually (but not necessarily) continuous, by using probability scores as the predicted values of the dependent variable.

Multilayer perceptron

A multilayer perceptron (MLP) is a feedforward artificial neural network model that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. Except for the input nodes, each node is a neuron (or processing element) with a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training the network [7][8]. MLP is a modification of the standard linear perceptron and can distinguish data that are not linearly separable [9].

PERFORMANCE RESULTS

All the data used in this experiment were collected into one table in the database. This table has 73 columns for process attributes and Defect Percentage Class for the output of the model. All algorithms were tested by Weka on a tenfold testing method. Major keypoints that were analyzed in the study are Accuracy rate and Kappa value. Having a high value in both values means the model works fine. Landis and Koch (1977) suggest that a kappa score over 0.4 indicates a reasonable agreement beyond chance.

Accuracy = (number of correct classifications given by the system) / (total number of test instances)

$K = (P_0 - P_c) / (1 - P_c)$

P0 is the accuracy of the classifier and Pc is the expected accuracy.

Algorithm	**Kappa**	**Accuracy**
J48	0.67	79.47
Naive Bayes	0.45	67.54
KNN	**0.68**	**80.06**
Support Vector Machine	0.59	74.67
Logistic Regression	0.56	73.56
Multilayer Perceptron	0.66	79.47

Table 4: Results of Algorithms.

CONCLUSION

Data of a real-world textile dyeing factory were tested by six different classification algorithms. As seen in the result table 4, accuracy and kappa statistics values of the experiments of KNN are slightly higher than the others. There were many noisy data in the dataset. After cleaning noisy data, the dataset was clear and had related data to analyze wastage amounts. Kappa statistics reveal that these results stem from a considerable classification success beyond chance. Attributes in real data are dependent on each other. According to the Naïve Bayes algorithm, attributes are not dependent on each other. Because of this, the kappa value of Naïve Bayes is lower than the others.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest regarding the publication of this paper.

REFERENCE

[1] Witten IH and Frank E. Data mining: Practical machine learning tools and technique, 2nd ed. Amsterdam: Elsevier Inc, 2005.

[2] Jiavei Han, Micheline Kamber, Jian Pei: Data Mining Concepts ans Techniques.

[3] Peter Flach: Machine Learning The art and science of algorithms that make sense of data.
[4] Ian H. Witten, Eibe Frank: Data Mining Practical Machine Learning Tools and Techniques.
[5] Ethem Alpaydın: Introduction to Machine Learning (Adaptive Computation and Machine Learning series).
[6] Clinical Research for Surgeons. Mohit Bhandari,Anders Joensson. p. 293.
[7] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.
[8] Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundations. MIT Press, 1986.
[9] Cybenko, G. 1989. Approximation by superpositions of a sigmoidal function Mathematics of Control, Signals, and Systems, 2(4), pp. 303–314.
[10] Yılmaz Kılıçaslan, Edir Serdar Güner, Savaş Yıldırım: Learning-based pronoun resolution for Turkish with a comparative evaluation
[11] Beatriz Reutter, Paul Lant, Christian Reynolds, Joe Lane: Food waste consequences: Environmentally extended input-output as a framework for analysis
[12] Watt Kwong Wai, Lin Ting, Liang Wei Pang, Hadianto Budihardjo, Gan Chiu Liang, Ding Liya, Zhu Fangming, Charles Pang T-Howe: Decision Support System for Production Scheduling (DSSPS)
[13] M.C. Barma, R. Saidur, S.M.A. Rahmand, A. Allouhi, B.A. Akash, Sadiq M. Sait: A review on boilers energy use, energy savings, and emissions reductions
[14] Jorge Artur Pec¸anha de Miranda Coelho, Valdiney Veloso Gouveia, Gustavo Henrique Silva de Souza, Taciano Lemos Milfont, Bruna Nogueira Romariz Barros: Emotions toward water consumption: Conservation and wastage
[15] Māris Jurušs, Elza Seile: Application of Loss Rates for Petroleum Products Due to Natural Wastage in Customs Procedures