Comparison of Machine Learning Algorithms to Build a Predictive Model for Classification of Survey Write-in Responses

Andrea Roberson[1], Justin Nguyen[1]
[1]U.S. Census Bureau, 4600 Silver Hill Road, Washington, D.C. 20233-1912

Proceedings of the 2018 Federal Committee on Statistical Methodology (FCSM) Research Conference

Abstract
The Annual Capital Expenditures Survey (ACES) provides detailed and timely information on capital investment in structures and equipment by nonfarm businesses during the year. The data are used to improve the quality of current economic indicators of business investments, as well as estimates of gross domestic product. Studies conducted by the U.S. Census Bureau have assessed Economic Directorate survey processing procedures and targeted areas for improvement. The resulting initiatives for the ACES questionnaire sought to reduce the workload of analysts who review and edit write-in responses. The form allows respondents to write-in a capital expenditure category as an "Other" category, when no other classification can be determined. This paper aims to review the use of machine learning to develop and validate predictive models to separate the write-ins in the "Other" category into multiple classes. We will examine how the Census SABLE (Scraping Assisted By LEarning) tool is applied to classify the "Other" category into multi-label descriptions: Structures, Equipment and Not Applicable.

Keywords: U.S. Census Bureau, machine learning, SABLE, predictive model

Section 1. Introduction

Machine learning is a type of artificial intelligence (AI) that enables software applications to more precisely predict outcomes, without being programmed explicitly. The fundamental proposition of machine learning is to develop algorithms that can take in data, and then utilize statistical methods to predict outcomes within sufficient bounds. The iterative aspect of machine learning is significant by virtue of the model's exposure to unseen data; this provides the ability for autonomous adaptation. AI algorithms learn from previous data to produce reliable, repeatable decisions and results (Liu et al., 2017). It is a science that is not brand-new, but one that has received modern momentum. An explosion of computing power is at the heart of modern machine learning, separating it from the past iterations of machine learning. While many machine learning algorithms have existed for decades, the power to automate the application of complex mathematical calculations to big data is novel. The production of fast and scalable machine learning algorithms is a recent advancement.

Section 2. Modernization of Statistical Production

2.1 ACES Research Overview

The Annual Capital Expenditures Survey (ACES) provides detailed and timely information on capital investment in structures and equipment by nonfarm businesses during the year. The data are used to improve the quality of current economic indicators of business investments, as well as estimates of gross domestic product. Figure 1 provides a snippet of the questionnaire. ACES survey Item 2 asks respondents to itemize their Expenditures into three classes. Survey respondents are asked to report dollar amounts for Structures in column 1, and Equipment in column 2. There is an option to allocate an amount in column 3 for items that have not adequately been described by the first two categories.

Any views expressed are those of the authors and not necessarily those of the U.S. Census Bureau.

Once that amount is entered, the respondent is directed in Item 3 to detail a written description of that Expenditure. Our goal was to examine the efficacy of using machine learning (ML) techniques to classify the written texts in the "Other" category into the descriptions: Structures, Equipment and Not Applicable.

Figure 1. Write-in classification processes

ITEM 2 **CAPITAL EXPENDITURES**

Report the following domestic capital expenditures data for the entire company. *(Refer to page 4 of Instructions)* **Example: if figure is $1,179,125,628.00 report →**

		Bil.	Mil.	Thou.
		1	1 7 9	1 2 6

Row	CAPITAL EXPENDITURES *(Refer to Page 2 of Instructions)*	Structures (1)			Equipment (2)			Other *(Describe in Item 3)* (3)			Total *(Add columns 1+2+3)* (4)		
		Bil.	Mil.	Thou.	Bil.	Mil.	Thou.	Bil.	Mil.	Thou.	Bil.	Mil.	Thou.
20	Capital expenditures for NEW structures and equipment *(Include major additions, alterations, and capitalized repairs to existing structures)*												
21	Capital expenditures for USED structures and equipment												
22	**TOTAL capital expenditures** *(Add Rows 20 + 21)*												
											Total should equal Item 1A, Row 11		

ITEM 3 **List the items included in "Other."** Report in thousands of dollars. **Furniture and fixtures, computers, capitalized computer software, and motor vehicles** should be reported as equipment. **Leasehold improvements** should be considered new structures or new equipment based on what is being improved.

Row	(1) Description of Capital Expenditures	(2) Bil.	Mil.	Thou.
30				

Source: 2017 Annual Capital Expenditures Survey, https://www.census.gov/programs-surveys/aces.html

Studies conducted by the U.S. Census Bureau have assessed Economic Directorate survey processing procedures and targeted areas for improvement. The resulting initiatives for the ACES questionnaire sought to reduce the workload of analysts who review and edit write-in responses.

In the recommendations of the Census Edit Reduction Team, it was suggested that ACES staff should consider making changes to address some of their most analyst burdensome edits, which included the classification of write-ins in the "Other" category. They believed this burden could be relieved through automation techniques such as ML. U.S. Census Bureau ML researchers are combining statistics and computer science to build algorithms that can solve our business needs more efficiently. This research is part of ongoing efforts to modernize statistical production by harnessing the potential of artificial intelligence.

2.2 Strategies for Modernization

Recently Statistics Canada conducted a study of the ML techniques currently in use or in consideration at statistical agencies worldwide (Chu and Poirier, 2015). This is an area in which many exciting advancements have been made over the past decade. Findings by Statistics Canada recommended that all National Statistical Offices explore the possibility of using ML techniques.

A wide variety of modeling strategies have been described in the literature. Decision trees have been used by Statistics Portugal. The purpose was to detect errors in foreign trade transaction data. The system was able to reduce by half the manual examination of records, while successfully detecting about 90% of the error-containing records. The Australian Bureau of Statistics (ABS) employed fully automatic categorization using support vector machines to code data from their 2006 Australian Census (Clarke and Brooker, 2011). A host of other ML strategies are currently in development world-wide. In this work we consider two closely linked strategies for classification, Support Vector Machines and Logistic Regression.

2.3 Support Vector Machines

Support Vector Machines (SVMs) were developed by Vapnik (2000) based on the structural risk minimization principle from statistical learning theory. Statistical Learning Theory, the backbone of SVMs, provides a new framework for modeling learning algorithms, merges the fields of ML and statistics, and inspires algorithms that overcome many theoretical and computational difficulties. In recent years, SVMs

has found a wide range of real-world applications, including face detection from images (Osuna et al., 1997; Shih and Liu, 1996), object recognition (Blanz et al., 1996; Hayasaka et al., 2006), speaker identification (Schmidt, 1996; Moreno and Ho, 2003), biomedical data classification (Shoker et al., 2005), and text categorization (Joachims,1997). The many applications of SVMs for text categorization generated considerable research interest for our study.

Joachims (2001) explains how SVMs can achieve good classification performance despite the high-dimensional feature spaces in text classification. The complexity of text-classification tasks are analyzed and sufficient conditions for good generalization performance are identified. The paper also provides a formal basis for developing new algorithms that are most appropriate in specific scenarios. The disadvantage of SVMs is that the classification result is purely dichotomous, and no probability of class membership is given (Masood and Al-Jumaily, 2013). Another disadvantage of SVMs is the black box nature of these functions.

2.4 Logistic Regression

Regression modeling is one of several statistical techniques that enable an analyst to predict a response based upon a set of inputs. Linear regression models are commonly used when the range of the response is continuous, and can theoretically take any value. LR, invented in the 19th century for the description of the growth of population and the course of chemical reactions, predicts the probability of an occurrence of an event by fitting data to a logistic curve (Zhang, Johnson, and Wang, 2012). As the output is restricted to the interval $(0, 1)$, the assumption of an infinite range fails. The logistic function used in this prediction method is useful in that it can take any value from negative infinity to positive infinity as input.

2.5 Comparison between Logistic Regression and Support Vector Machines

Logistic regression is a statistical model. Here the dependent variable is a category (Structures or Equipment). We have a set of text as predictors or features, which come from our survey write-in responses. This is called training data in ML terminology. The statistical maximum likelihood technique is used to find optimal values for the model parameters. The resulting model is used to predict the class of new survey write-ins.

Support Vector Machines, however, are non-probabilistic classifiers. It has the same goal as LR. Given training data, find the best SVM model, and use the model to classify new survey responses. The difference is that the optimization problem is finding the hyperplane that best separates the write-ins labeled "Structures" from those labeled "Equipment".

The research question considered in this work, is whether these two approaches, a traditional linear approach, and a newer, nonlinear approach can give us better insights into the classes that our ACES write-in data fall in.

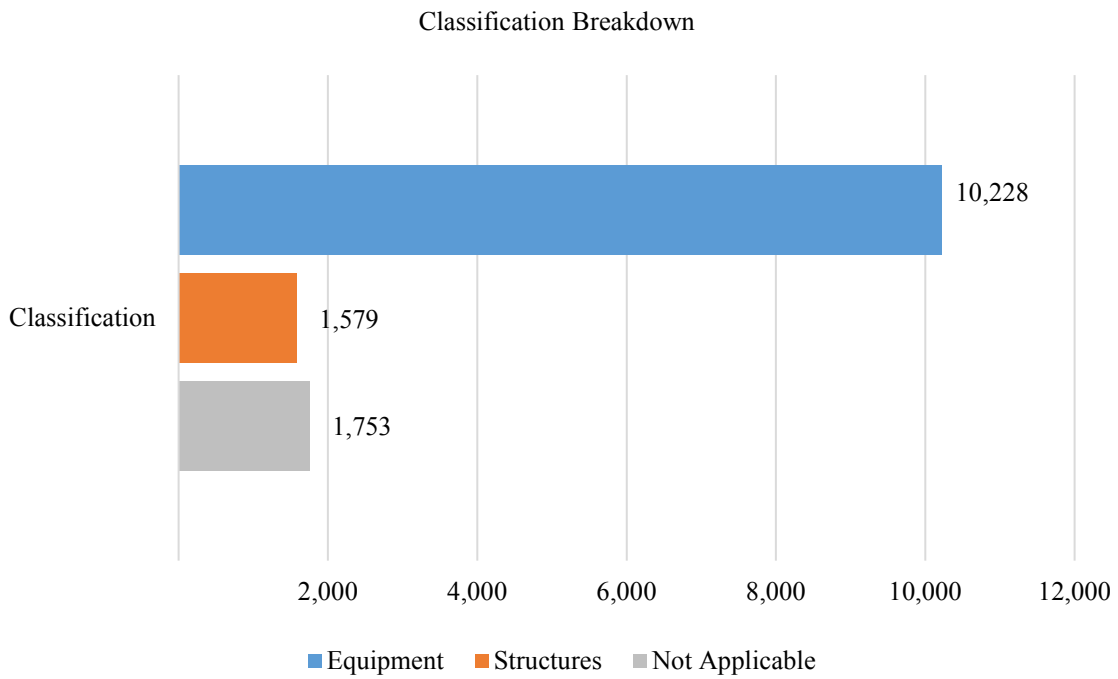
Section 3. Methodology

3.1 Data

The data used in this study was provided by ACES staff from data collection acquired in survey years 2015 and 2016. The classification goal is to predict if write-ins in the "Other" category are truly Structures, Equipment, or Not Applicable.

We have disproportionate class labels in the response variable. The total data set consisted of 13,560 labelled write-ins, of which 10,228 were labelled Equipment, 1,579 were labelled Structures, and 1,753 were labelled Not Applicable. The distribution of our write-in data is pictured in Figure 2.

Figure 2. Write-in classification processes



Source: U.S. Census Bureau, 2015 and 2016 Annual Capital Expenditures Survey

3.2 Steps for Text Analysis

The collected data were transformed to a structured format. These steps are commonly applied for information retrieval, information extraction and data mining. This was done by applying text processing techniques on the write-ins. Punctuations and numbers were removed from the write-in text. Next, all of the letters were converted to lowercase. Another common preprocessing step is the removal of white space. It is typically the result of all the left over spaces or tabs that were not removed along with the words that were deleted. All white space was removed.

A further preprocessing technique is the removal of stop words. They are words which are filtered out before or after processing of natural language data (text). Any group of words can be chosen as the stop words for a given purpose. Stop words are words that are so common in a language that their information value is almost zero, i.e., they do not carry significant information (Blair, 1979). We would not want these words taking up space in our database, or taking up valuable processing time. Some examples are “a”, “about”, “be”, “do”. Therefore, it is recommended to remove them before further analysis. In this work we remove them with the Natural Language Toolkit (NLTK) library in python.

Word (or n-gram) frequencies are typical units of analysis when working with text collections. The general term n-gram means ‘sequence of length n’. A three-word sequence is called a trigram, a sequence of two words is called a bigram, and a single word is called a unigram. It may come as a surprise that reducing a book to a list of word frequencies retains useful information, but this has been demonstrated in natural language processing (NLP) research. Treating texts as a list of word frequencies (a vector) also makes available a vast range of mathematical tools developed for studying and manipulating vectors.

Text feature extraction is the process of transforming what is essentially a list of words into a feature set that is usable by a classifier. In Bag-of-Words feature selection, the document is treated as an unordered list of words. Under this approach, words are ranked solely by their frequencies. In this case, the set of feature vectors can be considered as a matrix where each row is one instance and each column represents a word

found in any of the documents. Thus, each cell (i, j) represents the number of times a word appears in the text of the document. It can be noted that this model builds a $n \times m$ matrix where, for our work, n is the the number of write-ins and m is the number of words without repetition that appear in the n write-ins.

In our analysis, we were able to extract features by using an n-gram model to transform the data into feature vectors for use in our models. We gathered word frequencies (or term frequencies) associated with texts into a document-term matrix using the CountVectorizer class from the scikit-learn python package.

3.3 Implementing Machine Learning Algorithms

The most widely used library for implementing ML algorithms in Python is scikit-learn. This library is a Python module integrating a wide range of state-of-the-art ML. This package focuses on bringing ML to non-specialists using a general-purpose high-level language. Emphasis is put on ease of use, performance, documentation, and API consistency.

A well-fitted model should not just provide good prediction accuracy on the data it was fitted to, it should also generalize to data not yet seen. We can estimate this generalization accuracy with a technique called cross-validation. The simplest form of cross-validation is as follows: the data are separated into a training set and a test set. The algorithm is fit on the training set and the accuracy (e.g. the percent correctly classified) is evaluated on the test set, giving an estimate of how the fit generalizes.

All classifiers will have various parameters which can be tuned to obtain optimal performance. Tuning is performed for varying values of the tuning parameters, searching for those that give the best generalization accuracy (Guenther and Scholau, 2016). This can be done by choosing a small number of possible values to test for each parameter, and trying all possibilities on the grid of their combinations. This is known as a grid search. In the context of ML, hyperparameters are parameters whose values are set prior to the commencement of the learning process. In scikit-learn, hyperparameter tuning can be conveniently done with the GridSearchCV estimator. It takes as input an estimator (such as accuracy) and a set of candidate hyperparameters. Cross-validation scores are then computed for all hyperparameter combinations, in order to find the best one. In this research we tune the LR and SVMs with GridSearchCV.

For LR, we use the sklearn.linear_model.LogisticRegression package in this scikit-learn library.

Parameters are as follows:

•penalty: It specifies the norm used in penalization. It can be 'l1', or 'l2'. The default value is 'l2'.
•C: It is the inverse of the regularization strength. Smaller values specify stronger regularization.

We first observe that setting the parameter C is crucial as performance drops for inappropriate values of C. The LR regularization parameter was set in the range of ($C = 10^{-4}, 10^{-3}, \ldots, 10^{5}, 10^{6}$). A large C can lead to an overfit model, while a small C can lead to an underfit model. We used GridSearchCV with 5-fold cross-validation to tune C in this hyperparameter space.

The package used for SVM classification in the scikit-learn library is svm.SVC.

Parameters are as follows:

•C: It is the regularization parameter, C, of the error term.
•kernel: It specifies the kernel type to be used in the algorithm. It can be 'linear', 'poly', 'rbf', 'sigmoid', 'precomputed', or callable. The default value is 'rbf'.
•degree: It is the degree of the polynomial kernel function ('poly') and is ignored by all other kernels. The default value is 3.
•gamma: It is the kernel coefficient for 'rbf', 'poly', and 'sigmoid'. If $gamma$ is 'auto', then $\frac{1}{n}$ features will be used instead.

Training SVMs with a linear kernel is faster than with any other kernel. When you train a SVM with a linear kernel, you only need to optimize the C regularization parameter. When training with other kernels, you also need to optimize the $gamma$ parameter, which means that performing a grid search will usually take more time. Therefore, linear kernels are indeed very well suited for text-categorization. It should be kept in mind, however, that it is not the only solution and in some cases using another kernel might be better. The recommended approach for text classification is to try a linear kernel first, because of its advantages. An SVM with a linear kernel is similar to logistic regression. Therefore, in practice, the benefit of SVMs typically comes from using non-linear kernels to model non-linear decision boundaries. In this study, in an effort to get the best possible classification performance, it was of interest to try the other kernels to see if accuracy was improved.

We did a set of experiments with different kernel functions such as the linear, RBF, polynomial, and sigmoid in order to see the quality of generalization for each kernel function. Using sklearn's SVM implementation svm.SVC, we apply a grid-search to find the best pair $(C, gamma)$ for each kernel function using 5-fold cross-validation. In order to increase efficiency, we try exponentially growing sequences of $(C, gamma)$ to identify good parameters (C = $2^{-5}, 2^{-3}, \ldots, 2^{15}$; $gamma$ = $2^{-15}, 2^{-12}, \ldots, 2^{12}$). After the optimal $(C, gamma)$ is found, the training data is trained using the SVMs with different kernels and the best parameters to generate the final models. After testing our SVM algorithm with various kernel transformations, we identified the linear kernel as the most efficient kernel that resulted in the highest classification results.

3.4 Model Evaluation Metrics

We define classification accuracy as the percentage of write-ins for which the classification agreed with the known categories. A write-in whose fitted state differs from the ground truth label, is defined to be an error. Classification accuracy, false discovery rate, specificity, and sensitivity were used as performance metrics. Following Rueda and Diaz-Uriarte (2007), we used the confusion matrix defined in Table 1 to estimate rates. We also calculated Cohen's Kappa coefficient (Cohen, 1960) from the confusion matrices to measure agreement beyond chance between the fitted results and the ground truth data. Kappa values range between -1 (all write-ins incorrectly classified) and 1 (all write-ins correctly classified). A Kappa value equal to zero indicates a performance no better than random.

We determined the entries of the confusion matrix as outlined in Table 1. The entries in the confusion matrix have the following meaning in the context of this study: *Ee* is the number of correct predictions that a write-in is Equipment. *En* is the number of incorrect predictions that a write-in is Not Applicable, when in fact it is Equipment. *Es* is the number of incorrect predictions that a write-in is Structures, when in fact it is Equipment. *E.* represents the total number of write-ins that were truly Equipment. Another way of expressing this total of write-ins that were truly equipment, is from the summation of the terms $Ee + En + Es$. These totals are presented in the last column of the confusion matrices.

Table 1. A confusion matrix (Provost and Kohavi, 1998) contains information about actual and predicted classifications derived by a classification system. The confusion matrix used to calculate rates (Rueda and Diaz-Uriarte, 2007)

True Class	**Predicted Class**			
	equipment	structures	not applicable	Total
Equipment	Ee	Es	En	E.
Structures	Se	Ss	Sn	S.
Not Applicable	Ne	Ns	Nn	N.

The formulas used to calculate these statistics are defined below. To understand the statistics, it is helpful to refer to Table 1.

Correct Classification Rate

$$CCR = \frac{Ee + Nn + Ss}{E. + N. + S.}$$

False Discovery Rate

$$FDR = \frac{Es + En}{En + Nn + Sn + Es + Ns + Ss}$$

Specificity

$$Specificity = \frac{Ee}{Ee + Es + En}$$

Sensitivity

$$Sensitivity = \frac{Nn + Ss}{N. + S.}$$

Section 4. Results

The processing results of each algorithm are given in Tables 2 and 3. We determined the entries of the confusion matrix as outlined in Table 1 of Section 3.

Table 2. Confusion Matrix Results for Logistic Regression

True Class		Predicted Class			
		Equipment (e)	Structures (s)	Not Applicable (n)	Total
	Equipment (E)	1386	1	3	**1390**
	Structures (S)	7	194	7	**208**
	Not Applicable (N)	12	8	182	**202**
	Total	**1405**	**203**	**192**	**1800**

Table 3. Confusion Matrix Results for Support Vector Machines

True Class		Predicted Class			
		Equipment (e)	Structures (s)	Not Applicable (n)	Total
	Equipment (E)	1387	1	2	**1390**
	Structures (S)	7	193	8	**208**
	Not Applicable (N)	12	8	182	**202**
	Total	**1406**	**202**	**192**	**1800**

Table 4 summarizes the four performance statistics for predicting Equipment and Structures in the test data set. SVM and LR achieved almost an identical correct classification accuracy of 97.9%. LR and SVM tied for a false discovery rate of .76%. SVM and LR had the same specificity at over 99%. LR achieved the best sensitivity of 91.7%. The Cohen's Kappa coefficients range from .9432 to .9433. The Kappa values indicate that model results were not due to chance. LR slightly outperformed SVMs in terms of the metrics calculated in Table 4.

Table 4. The performance statistics for the compared methods on the test data, and Cohen's Kappa coefficient.

Model	SVMs	LR
Correct Classification Rate	.9789	.9789
False Discovery Rate	.0076	.0076
Specificity	.9978	.9978
Sensitivity	.9146	.9171
Cohen's Kappa coefficient	.9432	.9433

Section 5. Conclusions

The analytic methods were found to be roughly equivalent in terms of their classification ability as demonstrated by several performance measures. LR remains the clear choice when the primary goal of model development is to look for possible causal relationships between independent and dependent variables, and a modeler wishes to easily understand the effect of predictor variables on the outcome given that the model equation is also provided.

In this study, LR achieved the highest correct classification rate, specificity, sensitivity, and the lowest false discovery rate. Finally, LR had the highest Kappa value, indicating this model had higher chance-corrected agreement with the ground truth data than SVM. For this data set, LR slightly outperformed SVM on several measures. These findings give us confidence that predictions from a deployed LR model will be comparable to the results produced by survey analysts.

We have also demonstrated that a traditional linear approach can give as good, or better predictive accuracy than a newer, nonlinear approach in the case of detecting the classes that our ACES write-in data fall in. We hope these results further the study of text categorization, as this research continues to become an essential topic in the study of ML and artificial intelligence.

The short-term goal of this work is to deploy our LR model into production for the 2017 ACES survey year, which in addition to allowing us to predict the write-in class, provides a probability associated with the prediction. This will be useful to both analysts reviewing our predictions, and future performance assessments. We also expect to receive more training data in the future that we want to be able to incorporate quickly into our model. Thus, we can propose that using LR will be more efficient. This modeling will improve operating performance by harnessing the power of Census data to make intelligent predictions.

Section 6. Future Research

A question we should address is whether it is possible to reduce the total number of features in the dataset. One approach is to use principal components analysis to reduce the number of features. Another approach is to use LR to select features and then use those features in an SVM. Studying the effects of the different features on the classification rate is out of scope of this paper. However, the design of good features is an important component for successful classification. Therefore, it is an important direction for future work.

Though empirical studies have shown that it is difficult to decide which metric to use for different problems, each approach has specific features that measure various aspects of the algorithms being evaluated (Li et al., 2009). It is often difficult to state which metrics are the most suitable to evaluate an algorithm (Aftarczuk, 2007). Assessing the performance of ML algorithms based on predictive accuracy, is often inappropriate in the case of imbalanced data. Chawla (2002) showed that a combination of over-sampling the minority class and

under-sampling the majority class can achieve better classifier performance (in ROC space) than only under-sampling the majority class. Other approaches to the construction of classifiers from imbalanced datasets should be considered, along with an evaluation of performance measures.

Acknowledgements
We would like to thank Carol Caldwell, Carma Ray Hogue, and Valerie Mastalski of the U.S. Census Bureau for reviewing this paper and providing helpful comments. Thanks also to Brian Dumbacher for his help and support regarding SABLE.

References

Aftarczuk, Kamila. (2018*). Evaluation of selected data mining algorithms implemented in Medical Decision Support Systems.* 10.21125/inted.2016.0487.

Blair, D. C. (1979). Information Retrieval. *Journal of the American Society for Information Science* 30(6), 374-375.

Blanz, V., Schölkopf, B., Bülthoff, H., Burges, C., Vapnik, V. and Vetter, T. Comparison of view–based object recognition algorithms using realistic 3d models. *Artificial Neural Network* ICANN'96, Berlin, 1996. Springer Lecture Notes in Computer Science, Vol. 1112, 251–256.

Chawla, Nitesh & Bowyer, Kevin & O. Hall, Lawrence & Philip Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Over-Sampling Technique. *Journal of Artificial Intelligence Research* 16, 321–357.

Clarke, F.R. and S.J. Brooker (2011). Use of ML for Automated Survey Coding. In *Proceedings of the 58th ISI World Statistics Congress*. August 21–26, 2011, Dublin, Ireland.

Cohen, J. (1960). A coefficient of agreement for nominal scales. *Educational and Psychological Measurement, 20,* 37-46.

Chu, Kenneth and Poirier, Claude (2015). Machine Learning Documentation Initiative, *CONFERENCE OF EUROPEAN STATISTICIANS.*

Guenther, N & Schonlau, M. (2016). *Support vector machines*. Stata Journal. 16. 917-937.

Joachims, T. (1997). Text categorization with support vector machines. Technical Report, LS VIII, no. 23. University of Dortmund.

Joachims, Thorsten. (2001). A Statistical Learning Model of Text Classification for Support Vector Machines. SIGIR Forum (ACM Special Interest Group on Information Retrieval). 128-136.

Li, X., Nsofor, G.C. and Song, L. (2009). A comparative analysis of predictive data mining techniques. *International Journal of Rapid Manufacturing*. v1 i2. 150-172.

Yue Liu, Tianlu Zhao, Wangwei Ju, and Siqi Shi. (2017). Materials discovery and design using ML, *Journal of Materiomics*, Volume 3, Issue 3, Pages 159-177.

Osuna, E., Freund, F. and Girosi, F. (1997). An improved training algorithm of Support Vector Machines. *Proc. IEEE Workshop Neural Networks for Signal Processing*.

Provost F., Kohavi R. (1998). Guest editors' introduction: On applied research in ML. *Machine Learning* 30(2–3): 127–132.

Rueda OM, Diaz-Uriarte R. (2007). Flexible and accurate detection of genomic copy-number changes from aCGH. *Computational Biology*, 3(6):1115-1122.

Schmidt, M. (1996). Identifying speaker with support vector networks. In *Proceedings of Interface*. Sydney.

Shih, P. and Liu, C. (2006). Face detection using discriminating feature analysis and Support Vector Machine. *Pattern Recognition*. Number 2, 260-276.

Shoker, L., Sanei, S. and Chambers, J. (2005). Artifact removal from electroencephalograms using a hybrid BSS-SVM algorithm. *IEEE Signal Processing Letters.* 12(10), 721-724.

Vapnik, V. and Chapelle, O. (2000). Bounds on Error Expectation for Support Vector Machines. *Neural Computation*. Volume. 12 Issue 9, 2013-2036.

Zhang, M., Johnson, G., and Wang, J. (2011). Predicting Takeover Success Using ML Techniques. *Journal of Business & Economics Research* (JBER).