Tension in big data using machine learning: Analysis and applications

Huamao Wang[a], Yumei Yao[b,*], Said Salhi[c]

[a] *The Global Centre for Banking and Financial Innovation (GCBFI), Nottingham University Business School, University of Nottingham, Nottingham NG8 1BB, UK.*
[b] *Adam Smith Business School, University of Glasgow, Glasgow G12 8QQ, UK.*
[c] *Centre for Logistics and Heuristic Optimisation (CLHO), Kent Business School, University of Kent, Canterbury, Kent CT2 7FS, UK.*

Abstract

The access of machine learning techniques in popular programming languages and the exponentially expanding big data from social media, news, surveys, and markets provide exciting challenges and invaluable opportunities for organizations and individuals to explore implicit information for decision making. Nevertheless, the users of machine learning usually find that these sophisticated techniques could incur a high level of tensions caused by the selection of the appropriate size of the training data set among other factors. In this paper, we provide a systematic way of resolving such tensions by examining practical examples of predicting popularity and sentiment of posts on Twitter and Facebook, blogs on Mashable, news on Google and Yahoo, the US house survey, and Bitcoin prices. Interesting results show that for the

[☆] The authors are grateful to the three anonymous reviewers, the editor, Alessandro Merendino, and the managing editor, Maureen Meadows, for their helpful comments and advice. Their invaluable comments improved both the content and the presentation of the paper. The authors alone are responsible for any errors that may remain.

[*] Corresponding author.

Email addresses: huamao.wang@nottingham.ac.uk (Huamao Wang), y.yao.1@research.gla.ac.uk (Yumei Yao), s.salhi@kent.ac.uk (Said Salhi)

case of big data, using around 20% of the full sample often leads to a better prediction accuracy than opting for the full sample. Our conclusion is found to be consistent across a series of experiments. The managerial implication is that using more is not necessarily the best and users need to be cautious about such an important sensitivity as the simplistic approach may easily lead to inferior solutions with potentially detrimental consequences.

Keywords: Big data; Machine learning; Data size; Prediction accuracy; Social media

1. Introduction and tensions due to big data

Texts on the online posts of social media and news websites provide a potentially unlimited new source of data that can be meaningful for organizations, individuals, and society (George et al., 2014). This new source of data is particularly valuable where relevant data on social outcomes are missing, e.g., the popularity or sentiment of a post. An accurate prediction of popularity provides benefits to organizations (or individuals) for decision making, such as the selection of products to promote (or purchase), see, e.g., Wamba et al. (2015) and Economist (2017). To this end, machine learning yields predictions from large-scale text data and allows us to recover implicit information for making effective decisions (see, e.g., Hou et al., 2018; Olhede and Wolfe, 2018).

Machine learning algorithms are not only powerful in discovering implicit relationships within unconventional high-dimensional big data but also convenient to implement on personal computers. This is mainly due to the advances in computer technology (Wang et al., 2018) and the rapid development of many tool packages in popular statistics systems and programming languages like R and Python (Pedregosa et al., 2011). Although a deep understanding of these algorithms can be mathematically demanding, these algorithms are now made accessible to a wider audience.

However, an important issue that is usually overlooked by academics and especially practitioners[1] is the tension that could arise due to the sensitivity of the prediction accuracy of machine learning methods with respect to several factors including the size of the data that are utilized to train a machine learning model. Other factors that could also cause tensions include the choice of the technique, the fine-tuning of the algorithm parameters, sampling methods, data characteristics, and problem objectives (see, e.g., Varian, 2014; Mullainathan and Spiess, 2017; Corbett, 2018). An appropriate choice of these factors is a challenging and complex issue since it could be problem-and-data dependent. Informative attempts to address some of these concerns are carried out in the research area of meta-learning (Reif et al., 2012).

To address the challenge, in our research, we firstly deal with the tension caused by the size of the training data set. We then perform a series of robustness experiments to demonstrate the consistency of our suggestion across different situations using other potential factors that could also cause tensions. In addition, to generalize our analysis and provide robust results,

[1] The practitioners and academics refer to the professionals who apply machine learning methods to a broad range of areas including promotion advertisements, brand strategies, the choices of services and products, economic analysis, managing operations, text and sentiment analysis, sports predictions, and spatial analysis (see, e.g., Miller, 2014; Kayser and Blind, 2017; Jun et al., 2018).

two random sampling methods are also considered.

Admittedly, one may assume that the power of computing is no longer a limitation and therefore adopting a straightforward and crude approach that selects the best option by evaluating all possible combinations of the factors causing tensions would be the easiest way forward. Though this complete enumeration technique might be feasible in certain situations, in many real-life instances it is necessary to find an efficient and novel approach that limits the evaluation of unnecessary computations. Furthermore, it is also crucially important to stress that it is usually overlooked that machine learning techniques, which are heuristics in the wider sense, do not guarantee optimality and therefore ignoring this important issue could be vital to practitioners as the obtained solutions can be potentially suboptimal and hence inferior, see Salhi (2017).

Indeed, in many cases, the performance of machine learning is simply counter-intuitive. For instance, a larger size of training data set may not necessarily result in a more accurate prediction. This important observation raises the tension which is the result of the following question: what is the most suitable training size one should use and how do we guarantee, at least empirically, that a prediction from the suggested size is sufficiently accurate?

We aim to respond to this question by providing solutions to alleviate such tension. To the best of our knowledge, this is the first time that the challenging issue of tackling the tension is explored. Specifically, we illustrate our approach by examining four cases of popularity and sentiment prediction based on big data from social media, blogs, and news websites. To provide extra robustness to our approach, we extended our experiments to two entirely different sectors, namely, the US house market and Bitcoin price prediction.

Our analysis of prediction accuracy shows that some machine learning methods perform well under different sizes of training data, while the prediction performance of many others fluctuates largely with the training size. The relative advantages of different methods vary with different data sets. Thus, the user needs to perform a systematic comparison of representative methods under a variety of training sizes. This comparison is practical if the computational time is not considerably high but it could become a burden otherwise. We solve this challenge by suggesting a generally suitable fraction of the full sample. According to our robust empirical results, using approximately 20% of the full sample often provides relatively significant gains in the accuracy compared with using the full sample.

The implicit premise for advocating big data is that better predictions or decisions will be achieved if more training data are available. In other words, the more the better. However, previous works point out that this is not always the case as more data could carry errors that lead to overfitting issues and misleading biases.

First, big data usually carry measurement errors, outliers, dependencies, and incidental homogeneity. For instance, Fan et al. (2014) discover that these errors make it difficult to validate many exogenous assumptions and result in wrong predictions and decisions. Lazer et al. (2014) also demonstrate that big data from Google Flu Trends lead to large errors in predictions. Hashem et al. (2015) summarize interesting challenges such as scalability, integrity, and heterogeneity while Hák et al. (2016) emphasize that the datadriven priorities of policy agenda may result in distortions.

Second, the overfitting issue widely exists in a variety of data sources. Hippert et al. (2005) address overfitting in the prediction of electricity load using neural networks. Liu and Gillies (2016) deal with overfitting related to feature extraction for classification tasks. Silva et al. (2018) examine overfitting in a semi-supervised learning problem where there is a lack of labeled data. Jollans et al. (2019) illustrate overfitting in neuroimaging data due to the characteristics of low signal-to-noise and large feature sets. Ha et al. (2019) show that overfitting in some models can be prevented by randomly dropping out some parts of the models during the iterative training process. Very recently, Abdollahi et al. (2020) discover that by allowing their model to remove randomly approximately 20% of the network nodes can prevent overfitting. We found that this is an exciting area of research that needs to be pursued.

Third, a wealth of information diverts the attention of an algorithm towards large biases. This is demonstrated by Corbett (2018) who highlights that more data bring more cognitive burden that produces more chances to make the biases persist or more acute. For example, in the case of short prediction, using a smaller but relevant amount of current information can be more useful than relying on a long time period where a large amount of information is irrelevant and likely to distort the results. This idea was also noted by Cai et al. (2018) who demonstrate that removing irrelevant and redundant data from high-dimensional data is necessary to improve learning accuracy. Similarly, de Amorim (2019) proposes a method to deal with only a fraction of a data set by removing irrelevant features. Jollans et al. (2019) point out that the data points in neuroimaging data are more than subjects, which affects the performance of prediction methods. Kanarachos et al. (2019) show that instantaneous monitor of vehicle fuel consumption using smartphones leads to rich data causing prediction models' performance losses, whereas ignoring GPS speed or acceleration could improve performance.

Our paper relates to the recent literature that studies the development and applications of big data and machine learning in social science. Kayser and Blind (2017) address the use of new textual data sources and prediction methods in new research questions. Blazquez and Domenech (2018) propose an architecture with new data sources to predict the changes in social and economic agents. Jun et al. (2018) review a wide range of research areas that employ big data to overcome the restriction of a single source. Iqbal et al. (2020) discuss the application in developing smart city including intelligent transportation. Gan et al. (2020) use deep learning technologies to provide an effective pricing method for Asian options in financial markets. In this study, we point out the challenges arising from a large number of heterogeneous sources of text data on social media and news websites.

The paper is organized as follows. Section 2 briefly describes the machine learning methods, data sets, and the performance measures that are used in this study. Sections 3 and 4 illustrate the tensions by investigating four scenarios for predicting the popularity and sentiment of posts on social media, blogs, and online news platforms. Section 5 describes the robustness of our findings under random sampling while Section 6 deals with the extension of our implementation to two additional experiments. In Section 7, we discuss data characteristics and managerial implications for different problem objectives. Section 8 summarizes our findings and suggestions. For completeness, we have also added appendices to show detailed results regarding the consistency of our findings under different sampling methods, prediction methods, and corresponding parameters.

2. Machine learning methods, data sets, and performance measures

In this section, we firstly provide a concise introduction to machine learning methods that we examine. We then present the data sets in social media and markets that are used as the platform to test our approach alongside the measures of performance that we adopted.

2.1. A brief on the machine learning techniques used

Machine learning is attracting a great amount of interest because it offers new tools to reveal generalizable patterns successfully from very complex and unregulated big data. The reason is that the assumptions in classical statistical techniques about the underlying structure are not considered necessary anymore. Machine learning achieves such a breakthrough since it turns the deductive problem of finding a rule to an inductive one by letting the data inform us the best rule characterizing data (Mullainathan and Spiess, 2017).

We briefly introduce those machine learning methods that we examined. We select four regression methods, namely, elastic net, gradient boosting, decision tree, and random forest. The four classification methods that we adopt include ridge, SVM (Support Vector Machines) with SGD (Stochastic Gradient Descent), decision tree, and random forest. These machine learning methods are easily available in the tool package *scikit-learn* of Python programming language developed by Pedregosa et al. (2011). These methods are purposely selected to (a) represent several widely-used families of algorithms and (b) demonstrate distinctive prediction accuracies. For completeness and to ensure that our chosen methods generally cover the possible range of accuracies, we provide a large number of methods with their respective accuracies in the appendices.

The algorithms discussed in this study are referred to as the "supervised" machine learning algorithms that solve a prediction problem. We do not explore "unsupervised" learning for pattern recognition and cutting-edge algorithms. Instead, we focus on the tensions in big data when we use supervised machine learning techniques with the aim of providing practical advice and examples for the applications of machine learning arising in the social sciences.

In our regression analysis, we use the "elastic nets" regressor, which applies a convex combination of Lasso (least absolute shrinkage and selection operator) with the L_1-norm of the coefficient vector as the regularizer and the ridge regressor with the L_2-norm. The regularity is imposed through a parameter α. Such a combination trains a sparse model like Lasso and meanwhile maintains the regularization properties of the ridge method. The elastic net method is particularly suitable to fit the data with multiple correlated features. Another linear method of doing text analysis and natural language processing is the SGD method. It fits linear models in a simple and efficient way by using different convex loss functions and different penalties. This is particularly appropriate for the data with a large number of observations and features. We use a loss function that fits a linear support vector machine (SVM).

The "random forest" algorithm is a widely-used technique to improve the prediction performance, see, e.g., Blazquez and Domenech (2018), Cai et al. (2018), and Jollans et al. (2019). It takes a weighted average over a number of so-called "decision tree" predictors that are restricted to a randomly selected subset of features. The "gradient boosting" algorithm builds an additive tree model by optimizing differentiable loss functions on its negative gradient. It has the advantages of naturally dealing with heterogeneous features of data, high predictive power, and robustness to outliers, see Pedregosa et al. (2011). We also use the "decision tree" method where only one single tree model is fitted rather than a few tree models as in the random forest case.

For simplicity, the default values of the parameters in these methods, as encoded in the *scikit-learn* package, are used here. This is a commonly used assumption since the users of machine learning methods, especially in social science, often rely on these default settings. We do not investigate the effects of tuning the parameter values but instead, we focus on the tensions in big data that are caused by the choices of machine learning methods and training sizes. Meanwhile, we also briefly discuss the way that we change the value of the parameter α for the linear models and the number of estimators for the ensemble-based models like the random forest below.

To choose the value of parameter α, we perform an iterative search in a given set of possible values by carrying out a so-called cross-validation scheme with iterative fitting along a regularization path. For the ensemble-based methods, we set the number of estimators to 5 for the random forest method as well as the other ensemble methods that are listed in Appendix A. However, for the gradient boosting we keep the default parameter value, 100, given by the *scikit-learn* package for the number of estimators. In this way, we show that using a large number of estimators in an ensemble method usually do not improve performance substantially besides being too time-consuming. For this reason, we choose a relatively smaller number of estimators which is found to achieve an accurate prediction relatively faster.

2.2. Data sets and the measure of performance

We apply the aforementioned methods to four examples of big data from social media (Twitter and Facebook), online blogs (Mashable) and news (Google and Yahoo). The first three are taken to provide a prediction of the popularity of a post while the fourth presents a sentiment analysis of news. To conduct a robustness analysis, we also examine two additional cases, namely, the American Housing Survey (AHS) and Bitcoin. We perform a regression analysis on the first two examples and the AHS case. We run a classification analysis on the third and fourth examples, as well as the Bitcoin case.

For each case, we take a range of percentages (20%, 25%, ..., 100%) of the full sample to demonstrate the tension of big data in terms of the selection of data sizes. Within each selected percentage of the full sample, we examine four partitions of training and testing sets. For example, consider the 75%/25% partition. After taking 20% of the full sample as the observations, we further use 75% of these observations, i.e., 75% × 20% of the full sample, as the training set. Then the remaining 25% of the observations are allocated for the testing set to measure the prediction accuracy of a given machine learning model.

To measure the performance of regression, we use the variance score R^2, which shows the extent to which future observations are likely to be predicted by the model. The *scikit-learn* package provides the *r2_score* function to compute R^2, which is defined as

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=0}^{n_{samples}-1}(y_i - \hat{y})^2}{\sum_{i=0}^{n_{samples}-1}(y_i - \bar{y})^2},$$

where $\hat{y}_i$ is the predicted value of the i-th observation, y_i is the corresponding true value, and the sample mean $\bar{y} = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} y_i$. A value of 1 indicates the best model while 0 corresponds to a constant model that disregards the input features. To quantify the tension of determining the size of the training set, we compute the deviation (in %) of the variance score R^2 using a partial set against its corresponding full set as follows:

$$Deviation\, of\, R^2 = 100 \times \frac{R^2_{Partial} - R^2_{Full}}{|R^2_{Full}|}.$$

A positive percentage difference indicates that using a partial sample achieves a better performance than utilizing a full sample.

For the classification problem, the classification accuracy score is defined in the range of 0 to 1 to measure the prediction performance. The *scikit-learn* package provides the *accuracy_score* function to compute the classification accuracy as follows.

$$Accuracy(y, \hat{y}) = \frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbf{1}(\hat{y}_i = y_i),$$

where $\mathbf{1}(x)$ is the indicator function. Similarly, we compute the deviation (in %) of accuracy using a partial set against its corresponding full set as follows:

$$Deviation\,of\,Accuracy = 100 \times \frac{Accuracy_{Partial} - Accuracy_{Full}}{Accuracy_{Full}}.$$

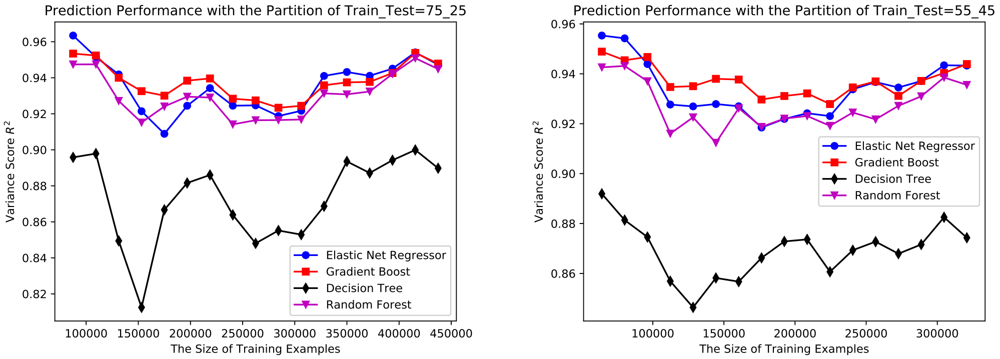
In short, machine learning algorithms generate boundless and exciting opportunities for organizations, individuals, and society in general for exploring helpful information in big data. Nevertheless, these algorithms raise tensions when they are trained with big data, which we aim to illustrate using applications in social media, news websites, and markets.

3. Tensions using big data: Applications to the prediction of popularity

Goes (2014) and Bello-Orgaz et al. (2016) summarize that the advantage of big data comes from the ability of machine learning to reveal information from various combinations of the "5Vs" characteristics: volume, variety, velocity, value, and veracity. Data from social media and news websites are typical big data with these characteristics. In this section, we examine the tensions arising from the applications of machine learning to the prediction of the popularity of posts on social media and news websites.

3.1. Prediction of Twitter post popularity

We start to illustrate the tensions in big data by examining a regression problem based on Twitter data. Usually, after a message is posted, there is a high volume of message exchange taking place in the day and the replies to the message largely occur during the following week. Due to the popularity and the fast-spreading speed of message, Twitter users' online activity data are therefore a natural source for analyzing popularity.



(a) Partition of 75% Training / 25% Testing (b) Partition of 55% Training / 45% Testing

Figure 1. Different Regression Methods for Twitter Data.

Figure 1 plots the regression variance scores of four machine learning methods under (a) the partition of 75% observations as the training set and 25% observations as the testing set; (b) the partition of 55% observations as the training set and 45% observations as the testing set.

We use the Twitter users' activity data that are collected by the AMA (Data Analysis, Modeling and Machine Learning Group) website of Grenoble Informatics Laboratory, Joseph Fourier University and 'BestofMedia' Group, see Mayuri et al. (2015).[2] We apply a group of machine learning regression methods for predicting the popularity of a post during the time-series of the data. The popularity is measured by the mean number of active discussions (NAD). There are 583,250 instances (i.e. observations), where each instance comprises 77 features (i.e. variables) that record the evolution of these features through the time series of the data.

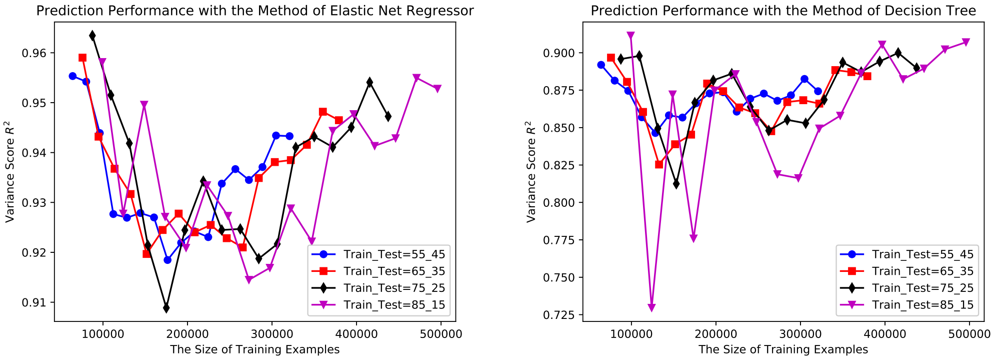
Figure 1 plots the changes in the regression variance scores of four machine

[2] http://ama.liglab.fr/resourcestools/datasets/buzz-prediction-in-social-media/

learning methods in accordance with the changes in the size of the training set. To vary the size of the training set, we incrementally take 20% to 100% of the full sample as the observations. For each size of the observations, Figure 1(a) splits the observations into a partition of 75% observations as the training set and 25% observations as the testing set. Similarly, Figure 1(b) shows the results based on the partition of 55% observations as the training set and 45% observations as the testing set.

Figure 1 implies two types of tensions that the user of machine learning will encounter. The first tension is that the user has to choose appropriate methods for the data set. To avoid selecting those methods that perform worse, one way would be to perform a comparison of different methods and choose the overall best solution. This strategy requires the user to perform this comparison for each data set because a method that is superior for a given set of data might not be so for the other data sets. In other words, there is no one method that guarantees to outperform every single one under all conditions as these techniques are heuristics in nature (Salhi, 2017). If this was not the case, such a method will be referred to as an optimal or "exact" method making the use of other methods obsolete. Figure 1 shows that the performance of the decision tree method is worse than the performances of the other three methods whose performances happen to be close to each other and fluctuate slightly. For example, in Figure 1(a) the R^2 of the elastic net regressor is lower than the gradient boost regressor when the size of the training set is 174,975, while the former performs better when the size is 349,950.

The second tension in the big data is that the prediction performance of



(a) Elastic Net Method

(b) Decision Tree Method

Figure 2. Different Partitions of Twitter Data for Regression.

Figure 2 displays the regression variance scores using (a) the Elastic Net Method; (b) the Decision Tree Method, under four partitions of observations with the ratios of the training set to the testing set as: 55%/45%, 65%/35%, 75%/25%, and 85%/15%.

a model does not necessarily improve after more observations are provided to train the model. Figure 1(a) shows that the four methods perform relatively worse when the sizes of training observations are around from 150,000 to 200,000 and from 250,000 to 300,000. On the contrary, both Figure 1(a) and Figure 1(b) show that under both partitions of 75%/25% and 55%/45%, the four methods all perform relatively better with 20% of the sample (87,487 observations) when compared to using the full sample.

To highlight the second tension about the size of the training set, Figure 2 depicts the changes in the variance score R^2 with respect to the size of the training set for the four partitions of data with the ratios of the training set to the testing set as: 55%/45%, 65%/35%, 75%/25%, and 85%/15%. Figure 2(a) uses the elastic net method and the R^2 scores are at the bottom when the size of the training set is around 200,000 to 300,000. Figure 2(b) plots

Table 1. Deviation of Variance Score R^2 w.r.t. the Full Sample (%) using the Partition of Train/Test=75/25 for Twitter Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	87,487	131,231	174,975	218,718	262,462	306,206	349,950	393,693
Elastic Net	1.71	-0.57	-4.05	-1.37	-2.38	-2.70	-0.43	-0.23
Gradient Boost	0.58	-0.83	-1.88	-0.87	-2.16	-2.47	-1.10	-0.55
Decision Tree	0.68	-4.53	-2.59	-0.42	-4.69	-4.14	0.42	0.50
Random Forest	0.27	-1.88	-2.20	-1.67	-3.01	-2.97	-1.49	-0.28

Notes. The first row shows the fraction of the observations with respect to the full sample. The last four rows display the deviations (in %) of R^2 where a positive value indicates a better performance than the full sample.

the results of the decision tree. The results fluctuate largely for the training size below 200,000 and gradually turn to be more stable for larger sizes of the training set. These figures all show that the input with 20% of the full sample leads to a better performance than the input of the full sample.

Table 1 displays the deviations of R^2 for the four methods under the partition of 75%/25%. The results show that the elastic net method obtains more than 1% gains, followed by the gradient boost and the decision tree that get more than 0.5% gains, whereas the random forest generates the least gains. As the elastic net method performs better than the others, we provide in Table 2 the corresponding deviations of R^2 for the following four partitions of the data, namely, 55%/45%, 65%/35%, 75%/25% and 85%/15%. Table 2 indicates that using 85% input data as the training set generally performs worse than other partitions using smaller inputs as the training sets. In summary, Table 1 and Table 2 indicate that using the four models with the

Table 2. Deviation of Variance Score R^2 w.r.t. the Full Sample (%) using the Method of Elastic Net for Twitter Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	64,157	96,236	128,315	160,393	192,472	224,551	256,630	288,708
Train/Test=55/45	1.28	0.07	-1.73	-1.73	-2.27	-2.15	-0.70	-0.66
Training Size	75,822	113,733	151,645	189,556	227,467	265,378	303,290	341,201
Train/Test=65/35	1.32	-1.03	-2.83	-1.98	-2.22	-2.69	-0.89	-0.52
Training Size	87,487	131,231	174,975	218,718	262,462	306,206	349,950	393,693
Train/Test=75/25	1.71	-0.57	-4.05	-1.37	-2.38	-2.70	-0.43	-0.23
Training Size	99,152	148,728	198,305	247,881	297,457	347,033	396,610	446,186
Train/Test=85/15	0.56	-0.34	-3.35	-2.67	-3.77	-3.21	-0.53	-1.04

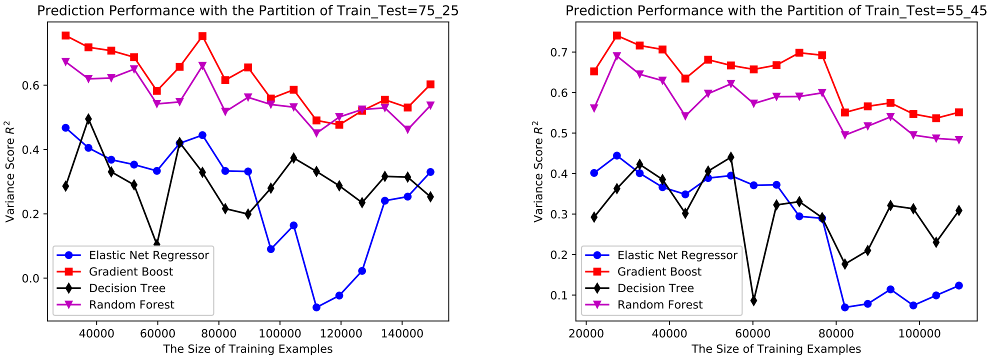
Notes. The first row shows the fraction of the observations with respect to the full sample. The last eight rows display the training sizes and the deviations (in %) of R^2 for four partitions of data: 55%/45%, 65%/35%, 75%/25%, and 85%/15%.

20% of the full sample achieves a better performance than the models using the full sample.

3.2. Prediction of Facebook post popularity

We discuss the tensions in big data by running regression methods to a data set from another popular social media, Facebook. The users can put a post of texts, photos, and multimedia, which can be shared with other users who can make comments to the post. The number of comments received by a post can be used as an index for measuring the popularity of the post.

We use a set of data about Facebook comment volume that are available from the UC Irvine Machine Learning Repository (Singh et al., 2015; Singh,



(a) Partition of 75% Training / 25% Testing (b) Partition of 55% Training / 45% Testing

Figure 3. Different Regression Methods for Facebook Data.

Figure 3 plots the regression variance scores of four machine learning methods under (a) the partition of 75% observations as the training set and 25% observations as the testing set; (b) the partition of 55% observations as the training set and 45% observations as the testing set.

2016)[3]. There are 199,030 instances in this data set including 53 features that are extracted from the posts on Facebook. We predict the number of comments that are received by a post through a group of machine learning methods.

Predicting the popularity of Facebook posts with machine learning methods also raises the same two tensions in big data as highlighted in the previous case.

(i) The choice of the method - First, the user is faced with the difficulty in determining appropriate methods for his/her data. Figure 3 plots the prediction performance of four machine learning methods in terms of variance

[3] https://archive.ics.uci.edu/ml/datasets/Facebook+Comment+Volume+Dataset

scores R^2 under the 75%/25% partition and the 55%/45% partition respectively. Under both partitions, the elastic net method and the decision tree method display poorer performance with all values of R^2 being below 0.5. In other words, the two methods cannot explain more than 50% of variations of the testing sets. By contrast, the gradient boost method and the random forest methods achieve variance scores higher than 0.5 almost for all training sizes in Figure 3(a) and Figure 3(b).

(ii) The effect of the size of the training set - As shown in Figure 3, the R^2 values of the four methods decrease with the number of observations. This result is even more marked especially with the elastic net method which shows a large decline when the training size is large at around 120,000. More importantly, Figure 3 shows that training these machine learning models with a fraction of the full sample can yield large gains compared with fitting these models with the full sample. These results are summarized in Table 3 and Table 4.

Table 3 displays that generally the four methods can achieve better gains in the prediction performance when we use 20% to 50% of the full sample. The elastic net, gradient boost, and random forest produce 25% to 40% gains in the variance score when a training set of 20% of the full sample is used only, which is a training size of 29,854. In contrast, using 70% to 90% of the full sample leads to significant losses in the prediction performance with up to 116% worse for the elastic net method.

As an example, we consider the method of random forest to investigate its prediction performance by using four partitions of training and testing sets. Table 4 summarizes these results. In almost all cases, random forest using

Table 3. Deviation of Variance Score R^2 w.r.t. the Full Sample (%) using the Partition of Train/Test=75/25 for Facebook Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	29,854	44,781	59,709	74,636	89,563	104,490	119,418	134,345
Elastic Net	**41.62**	11.54	1.03	**34.73**	0.47	-50.34	**-116.39**	-27.07
Gradient Boost	**25.18**	17.38	-3.37	**24.89**	8.73	-2.80	-20.88	-7.97
Decision Tree	13.12	**30.71**	-58.63	**30.14**	-21.22	47.77	13.60	25.15
Random Forest	**25.24**	15.88	0.89	**22.95**	4.70	-1.00	-6.70	-1.43

Notes. The first row shows the fraction of the observations, which are used by the methods, to the full sample. The last four rows display the deviations (in %) of R^2 and a positive value indicates a better performance than the full sample.

Table 4. Deviation of Variance Score R^2 w.r.t. the Full Sample (%) using the Method of Random Forest for Facebook Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	21,893	32,839	43,786	54,733	65,679	76,626	87,573	98,519
Train/Test=55/45	15.74	33.51	12.16	28.63	21.72	24.03	7.03	2.89
Training Size	25,873	38,810	51,747	64,684	77,621	90,558	103,495	116,432
Train/Test=65/35	9.27	21.53	7.57	19.12	17.81	11.30	-9.57	-5.79
Training Size	29,854	44,781	59,709	74,636	89,563	104,490	119,418	134,345
Train/Test=75/25	25.24	15.83	0.87	22.99	4.70	-0.92	-6.68	-1.47
Training Size	33,835	50,752	67,670	84,587	101,505	118,422	135,340	152,257
Train/Test=85/15	25.81	40.37	-11.65	29.79	21.73	30.64	7.77	34.97

Notes. The first row shows the fraction of the observations, which are used by the methods, to the full sample. The last eight rows display the training sizes and the deviations (in %) of R^2 for four partitions of data: 55%/45%, 65%/35%, 75%/25%, and 85%/15%.

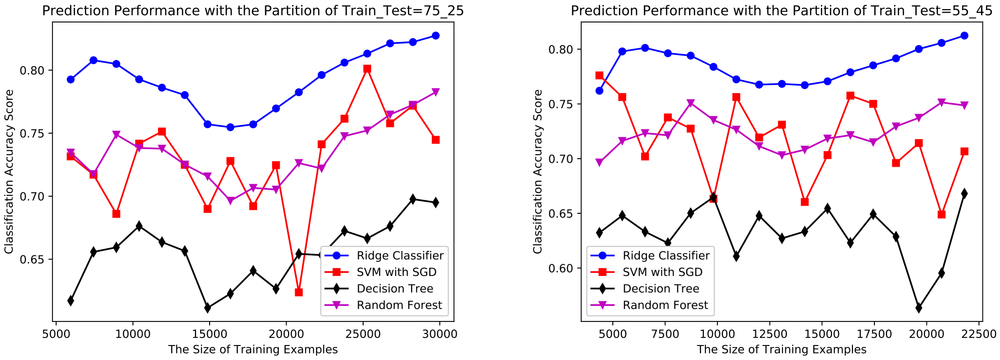
partial samples achieves a better performance than using the full sample. Besides, in most cases, the performance gain is around 20%. Also, the tension due to the sensitivity of the prediction performance to the training size strikingly stands out when we choose 40% of the full sample instead. In particular, the gain is over 12% under the partition of 55%/45% and it decreases sharply to 7.57% and 0.87% under the partitions of 65%/35% and 75%/25% respectively. The worst scenario is to adopt the partition of 85%/15%, which results in an important performance loss of 11.65% when compared with the full sample case. In other words, a user who happens to choose the 40% sample set with these chosen partitions could potentially incur significant performance losses.

3.3. Prediction of Mashable blog popularity

We now explore the case of classification problems using big data from a well-known online blog platform, namely Mashable. It publishes diverse topics and attracts millions of followers. We study a large data set with 39,000 articles published on the Mashable website, which is collected by Fernandes et al. (2015) and which can also be obtained from the UC Irvine Machine Learning Repository[4]. The data consist of 39,644 instances of articles, each containing 59 features.

We use a group of machine learning classification algorithms to predict the popularity of an article. The popularity of an article is measured by the number of shares of this article in some large social media platforms including Facebook, Twitter, Google+, LinkedIn, StumbleUpon, and Pinterest. We study a classification problem to distinguish between popular articles and those that are not. We do this by setting the mean of the number measuring

[4] https://archive.ics.uci.edu/ml/datasets/online+news+popularity



(a) Partition of 75% Training / 25% Testing (b) Partition of 55% Training / 45% Testing

Figure 4. Different Classification Methods for Mashable Data.

Figure 4 plots the classification accuracy scores of four machine learning methods under (a) the partition of 75% observations as the training set and 25% observations as the testing set; (b) the partition of 55% observations as the training set and 45% observations as the testing set.

popularity as the threshold.

We demonstrate the results of four classification methods of predicting the popularity of the blogs in the data set. Figure 4 highlights two tensions in big data that arise when users try to classify the blogs to find out whether they are popular. First, the classification accuracy scores demonstrate large differences for the methods of decision tree, random forest, and ridge classifier, as shown in Figure 4(a) and Figure 4(b). For example, a user who chooses the decision tree method will generally incur about 20% losses in accuracy scores compared with the one who opts for the ridge classifier.

Second, the four methods generally show a U-shape of accuracy scores when they are trained by using more training observations. For example, Figure 4(a) displays that a user taking about 15,000 to 20,000 training ob-

Table 5. Deviation of Classification Accuracy w.r.t. the Full Sample (%) using the Partition of Train/Test=75/25 for Mashable Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	5,946	8,919	11,892	14,866	17,839	20,812	23,786	26,759
Ridge Classifier	-4.21	-2.72	-5	-8.51	-8.51	-5.43	-2.59	-0.74
SVM with SGD	-1.76	-7.89	0.89	-7.37	-7.06	-16.26	2.25	1.76
Decision Tree	-11.21	-5.12	-4.52	-12.03	-7.79	-5.87	-3.24	-2.70
Random Forest	-6.13	-4.31	-5.73	-8.54	-9.71	-7.19	-4.46	-2.27

Notes. The first row shows the fraction of the full sample used by the methods. The last four rows display the deviations (in %) of accuracy where a positive value indicates a better performance than the full sample.

servations (i.e., 50% - 70% of the full set) will achieve the least prediction performance. The tension is more significant when the user discovers that the SVM (Stochastic Vector Machine) method displays large fluctuations in its scores. For instance, it generates a score which is lower than the one obtained by the decision tree at about 20,000 but at around 25,000 the same method produces a much better result by overtaking the random forest. It can also be noted that in this class, the ridge classifier outperforms the other methods over all training sizes making it appropriate for the user to opt for.

According to Table 5, although the four methods with partial samples of the data set generally cause performance losses in the classification accuracy, there are exceptions of positive performance gains coming from the SVM method. For instance, these happen when using 40% or 80% of the full sample. These exceptions reflect the large fluctuations in the performance of the SVM method as shown in Figure 4. To examine this method further,

Table 6. Deviation of Classification Accuracy w.r.t. the Full Sample (%) using the Method of SVM with SGD for Mashable Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	4,360	6,541	8,721	10,902	13,082	15,262	17,443	19,623
Train/Test=55/45	9.83	-0.66	2.95	7.03	3.46	-0.48	6.15	1.08
Training Size	5,153	7,730	10,307	12,884	15,460	18,037	20,614	23,191
Train/Test=65/35	-32.21	-15.20	16.21	-1.73	3.48	11.15	-4.19	11.70
Training Size	5,946	8,919	11,892	14,866	17,839	20,812	23,786	26,759
Train/Test=75/25	-1.76	-7.89	0.89	-7.37	-7.06	-16.26	2.25	1.76
Training Size	6,738	10,109	13,478	16,848	20,218	23,587	26,957	30,327
Train/Test=85/15	-4.13	-13.78	-16.15	-14.21	-19.81	-11.78	-12.80	-3.57

Notes. The first row shows the fraction of the full sample used by the methods. The last eight rows display the training sizes and the deviations (in %) of accuracy for four partitions of data: 55%/45%, 65%/35%, 75%/25%, and 85%/15%.

Table 6 displays the deviations of accuracy using partial samples with respect to the accuracy with the full sample under four different partitions. We find that using the SVM method with partial samples under the training/test partitions of 55%/45% and 65%/35% have more chances to outperform the method with the full sample. Therefore, the tension in big data caused by the sensitivity of classification methods to the training size is still serious when the SVM method is applied to the Mashable data.

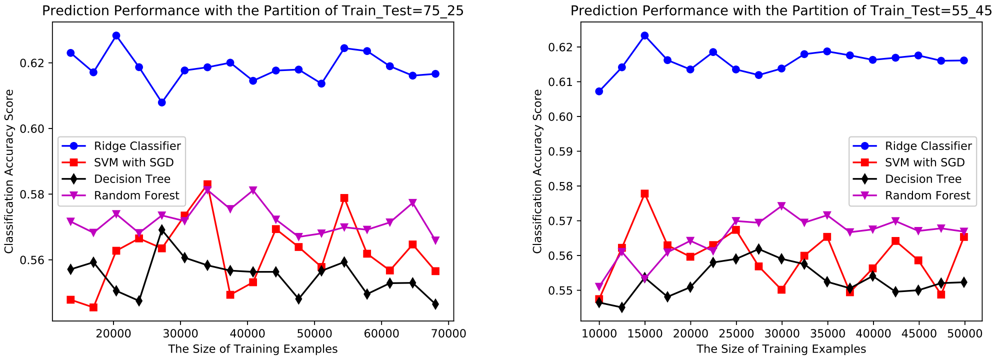
4. Tensions using big data: Application to the prediction of news sentiment

In this section, we illustrate the tensions in big data from Online News when a user wants to classify the news into two categories, namely, the optimistic and the pessimistic.

We study sentiment that is revealed by news on Google News and Yahoo News. We use a large data set of news items published on Google and Yahoo, which are collected by Moniz and Torgo (2018) and which can also be downloaded from the UC Irvine Machine Learning Repository[5]. The data carry the topics of news involving the economy, computers, and politics. The data also include the respective feedback of news on multiple social media platforms including Facebook, Google+, and LinkedIn. In addition, it provides a sentiment score derived from the title texts of the news items. The data contain 90,722 instances of news and there are 171 features in each instance. In this classification problem, we set the mean of the sentiment score as the threshold when we want to distinguish between optimistic news and pessimistic news.

Figure 5 shows that the ridge classifier outperforms the other three classification methods. It is also worth noting that the results obtained by SVM fluctuate around those found by the decision tree method and the random forest method, which makes the choice between these three methods difficult. A user who only tries the ridge classifier will not face the tension in the choice of the last three methods as the chosen method outperforms the others. Though the user has chosen the right approach on this occasion, he/she has to deal with the tension about the determination of an appropriate training size. As shown in Figure 5(a) and Figure 5(b), the ridge classifier with the input of a 30% of the full sample achieves higher accuracy scores than if

[5] https://archive.ics.uci.edu/ml/datasets/News+Popularity+in+Multiple+Social +Media+Platforms



(a) Partition of 75% Training / 25% Testing (b) Partition of 55% Training / 45% Testing

Figure 5. Different Classification Methods for News Data.

Figure 5 plots the classification accuracy scores of four machine learning methods under (a) the partition of 75% observations as the training set and 25% observations as the testing set; (b) the partition of 55% observations as the training set and 45% observations as the testing set.

Table 7. Deviation of Classification Accuracy w.r.t. the Full Sample (%) using the Partition of Train/Test=75/25 for Online News Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	13,608	20,412	27,216	34,020	40,824	47,628	54,432	61,236
Ridge Classifier	1.03	1.89	-1.41	0.32	-0.34	0.21	1.27	0.38
SVM with SGD	-1.56	1.12	1.25	4.76	-0.61	1.32	4.01	0.04
Decision Tree	1.94	0.74	4.14	2.17	1.80	0.29	2.35	1.17
Random Forest	1.02	1.42	1.35	2.71	2.70	0.19	0.71	0.97

Notes. The first row shows the fraction of the observations, which are used by the methods, to the full sample. The last four rows display the deviations (in %) of accuracy and a positive value indicates a better performance than the full sample.

the full or almost the full sample is considered instead.

Table 8. Deviation of Classification Accuracy w.r.t. the Full Sample (%) using the Method of Ridge Classifier

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Training Size	9,979	14,968	19,958	24,948	29,938	34,927	39,917	44,906
Train/Test=55/45	-1.44	1.16	-0.42	-0.42	-0.38	0.42	0.03	0.23
Training Size	11,793	17,690	23,587	29,484	35,381	41,278	47,175	53,071
Train/Test=65/35	0.38	1.20	-0.62	-1.06	-0.40	0.34	0.09	0.23
Training Size	13,608	20,412	27,216	34,020	40,824	47,628	54,432	61,236
Train/Test=75/25	1.03	1.89	-1.41	0.32	-0.34	0.21	1.27	0.38
Training Size	15,422	23,133	30,844	38,556	46,268	53,979	61,690	69,401
Train/Test=85/15	1.24	5.43	0.52	2.10	0.65	2.26	2.16	1.35

Notes. The first row shows the fraction of the observations, which are used by the methods, to the full sample. The last eight rows display the training sizes and the deviations (in %) of accuracy for four partitions of data: 55%/45%, 65%/35%, 75%/25%, and 85%/15%.

A quantitative analysis of this tension is further presented in Table 7 and Table 8 where the results reveal that in most cases with the four methods and under the four partitions of the data set, the use of partial samples yields positive gains in the prediction performance. Many of these gains range from 1% to 2%, except for a handful of cases where the gains are more than 4%. Particularly, the decision tree method and the random forest method with the 75%/25% partition in Table 7 and the ridge classifier with the 85%/15% partition in Table 8 produce positive gains in accuracy for the different fractions of the full data sample. These results empirically demonstrate the tension that simply using the full sample is likely to lead to inferior outcomes with considerable losses in the prediction accuracy.

5. Robustness analysis using random sampling

In Sections 3 and 4, we focused on the tension caused by the sample size only. For simplicity, we did not introduce randomness into the train/test partitions and in the partial samples of the above four data sets. In addition to the sample size, as we mentioned earlier, tensions can also be caused by other factors such as sampling methods, prediction methods and corresponding parameters, data quality and characteristics, and the objectives and complexities of the problem.

Hence, in this section, we show the consistency of our findings using randomly selected samples. In the appendices, we examine our findings further by using different sampling methods, key parameters, and prediction methods. Under random sampling, we report means and standard errors and our results imply that taking a partial sample usually provides a significantly more accurate prediction than the full sample.[6]

Specifically, we show that our findings hold when we take two different random sampling methods. Among the four cases of big data from social media, we take the case about the prediction of Mashable blog popularity as an example. Similar results can be derived for the other data sets. In Section 6 and Appendix B, we also apply the sampling methods to two additional cases and obtain the same conclusion. We report the results with different numbers of random sampling for generality. Besides, these results are obtained under a *random* partition of 75% observations as the training

[6] We are grateful to three anonymous referees' helpful comments about robustness and extension.

Table 9. Deviation of Classification Accuracy w.r.t. the Full Sample (%) with Random Sampling 35 Times for Mashable Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	0.81	0.49	0.77	0.75	0.80	0.73	0.91	0.62
std.	0.14	0.15	0.11	0.10	0.10	0.09	0.07	0.09
SVM with SGD	-2.71	-2.45	-1.73	-1.37	0.29	-0.69	0.69	-0.17
std.	0.66	0.66	0.48	0.45	0.63	0.59	0.45	0.48
Decision Tree	0.22	0.05	0.61	0.71	0.86	0.86	0.81	0.60
std.	0.23	0.22	0.18	0.16	0.15	0.13	0.10	0.09
Random Forest	0.75	0.45	0.93	1.03	1.06	1.02	1.25	1.04
std.	0.20	0.14	0.14	0.11	0.11	0.11	0.09	0.09

Notes. The first row shows the fraction of the observations from the sample examined. The last eight rows display the means and standard errors of the deviations (in %) of accuracy. A positive mean value indicates a better performance from random sampling on average.

set and 25% observations as the testing set within a particular sample that is examined.

Table 9 reports the deviation of classification accuracy with respect to the full sample in percentage points. Within each column, we randomly take a specific fraction of all observations in the full sample 35 times. The three methods (the ridge classifier, the decision tree, and the random forest) with partial samples (especially 20% of the full sample) lead to a higher prediction performance than those with the full sample. The SVM (Support Vector Machines) with SGD (Stochastic Gradient Descent) also provides a better performance with a fraction of the observations than the full sample.

To show the superior performance of partial samples with random sam-

Table 10. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 20% Complete Sample 5 Times for Mashable Data (Total 50 Random Runs)

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	0.48	0.53	0.20	0.43	0.55	0.59	0.48	0.50
std.	0.20	0.17	0.11	0.16	0.17	0.19	0.15	0.15
SVM with SGD	2.45	2.04	2.07	2.10	2.11	2.39	2.47	2.36
std.	0.78	0.62	0.62	0.62	0.65	0.72	0.75	0.72
Decision Tree	0.53	0.27	0.13	0.17	0.43	0.29	0.66	0.49
std.	0.20	0.21	0.11	0.15	0.19	0.11	0.20	0.16
Random Forest	0.37	0.45	0.07	0.29	0.53	0.38	0.41	0.55
std.	0.22	0.16	0.09	0.18	0.17	0.14	0.13	0.18

Notes. The first row shows the fraction of the observations from the sample examined. The last eight rows display the means and standard errors of the deviations (in %) of accuracy. A positive mean value indicates a better performance from random sampling on average.

pling further, we use a *two-loop* random sampling method and list the results in Table 10. In the first loop, we randomly take a *subsample* with 20% observations from the complete sample set of the data. That is 20% of 39,644 observations and is a *subsample* with 7,929 observations. In the second loop, for each column, we randomly take a specific fraction of the *subsample*, e.g., a random 30% of the 7,929 observations in the column of "0.30". We take 5 random runs in the first loop and 10 random runs in the second loop, resulting in a total of 50 random runs. Table 10 exhibits that for the four methods, adopting 20% to 90% fractions of the five random 7,929 observations provides more accurate results than using all these observations. These results imply that for the five random *subsamples* where each *subsample* carries 20% of the complete sample, a further 20% fraction of these subsamples is also sufficient to obtain a better prediction accuracy.

6. Experiments with big data in two other sectors

In addition to the previous four cases of big data on social media, we examine another two data sets from two different sectors in the economy, namely, the US house market and the Bitcoin market. These additional experiments are performed to assess further the robustness of our findings and find out whether taking a partial sample is still useful and valid.

6.1. Big data of the U.S. house market

The American Housing Survey (AHS) 2013 national data contain rich information about individual housing units.[7] We apply a procedure like Mullainathan and Spiess (2017) to clean the data. As a result, we take 35,852 instances with 138 features from the survey to predict house values by using four regression methods in machine learning.

Table 11 reports the deviations of R^2 after a two-loop random sampling with a total 100 random runs, where a random partial sample for each column is taken from a random subsample with 20% observations in the complete sample. Except for a small number of negative deviations, the positive deviations imply that on average, using a fraction (e.g., 20% for the decision tree method) of a subsample with 20% of the complete sample in AHS survey

[7] https://www.census.gov/programs-surveys/ahs/data/2013.html

Table 11. Deviation of Variance Score R^2 (%) with Random Sampling 10 Times, Based on a Random 20% Complete Sample 10 Times for the U.S. House Data (Total 100 Random Runs)

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Elastic Net	-2.53	0.93	1.76	4.71	2.95	2.52	1.98	2.27
std.	2.58	1.80	0.73	1.82	1.37	0.97	1.02	0.80
Gradient Boost	-4.60	0.14	-0.23	5.01	4.50	2.73	3.49	1.36
std.	2.50	1.60	1.09	1.86	1.84	1.08	1.29	0.79
Decision Tree	4.33	0.50	1.12	3.39	2.36	1.66	2.05	3.34
std.	1.94	1.59	0.93	1.51	1.68	1.53	1.39	1.15
Random Forest	7.26	-6.47	-11.45	14.78	16.67	10.67	12.76	6.84
std.	9.61	11.39	9.83	9.36	7.46	4.96	7.77	4.14

Notes. The first row shows the fraction of the observations from the sample examined. The last eight rows display the means and standard errors of the deviations (in %) of R^2. A positive mean value indicates a better performance from random sampling on average.

data can significantly achieve superior performance in the prediction of house values.

6.2. The Bitcoin price prediction with news data

Another additional case is a combination of news data and the Bitcoin data from the Kaggle website. First, we take 15,000 news titles from a data set that contains 48% records from Breitbart, 28% records from CNN, and 24% records from other U.S. newsagents (Thompson, 2017).[8] Second, we take the time-series data of Bitcoin that is one of the most popular crypto-

[8] https://www.kaggle.com/snapcrack/all-the-news/version/1

Table 12. Deviation of Classification Accuracy w.r.t. the Full Sample (%) with Random Sampling 40 Times for Bitcoin Data

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	14.62	16.88	11.33	13.39	13.86	13.11	13.66	13.68
std.	2.83	2.76	2.21	1.65	1.35	1.47	1.36	1.12
SVM with SGD	5.13	4.36	0.15	-0.85	6.35	1.82	5.63	6.88
std.	3.20	2.57	2.18	2.63	2.46	1.65	1.73	1.61
Decision Tree	9.14	7.09	0.48	7.05	1.99	-0.97	1.42	0.36
std.	3.31	2.80	2.30	1.44	1.90	2.02	1.81	1.48
Random Forest	1.05	6.09	1.97	3.42	1.56	0.18	3.03	-0.77
std.	2.83	2.49	2.08	2.25	1.53	1.81	1.38	1.46

Notes. The first row shows the fraction of the observations from the sample examined. The last eight rows display the means and standard errors of the deviations (in %) of accuracy. A positive mean value indicates a better performance from random sampling on average.

currencies.[9] Third, we employ the Term Frequency-Inverse Document Frequency (TF-IDF) technique to transfer the news titles to a large vector of word-based data and then merge them to the Bitcoin data according to the same dates.

We use the news data to predict the up or down trends of Bitcoin prices. Table 12 is obtained by simply taking 40 random partial samples for each column based on the complete sample. More robustness results with the two-loop random sampling are listed in Appendix B. Table 12 shows that the ridge classifier significantly achieves the largest performance gains in predic-

[9] https://www.kaggle.com/jessevent/all-crypto-currencies

tion when random partial samples are adopted. Using partial samples, the decision tree method and the SVM method usually provide a better prediction performance than those with the complete sample. The random forest method with partial samples also obtains performance gains, although the gains are relatively small compared with the other three methods. Indeed, Table 14 to Table 17 in Appendix B, which adopt partial samples that are randomly taken from the 20% to 80% of the complete sample, consistently demonstrate a superior performance. In addition, Table 18 to Table 20 show that this conclusion still holds for other alternative machine learning techniques and for different values of some key parameters in selected techniques.

Overall, our results from the two additional cases of big data again support that the tension due to the sample size is crucial in the data from social media, news, economics, and financial time-series data. Applying machine learning to a partial sample such as 20% of the full sample empirically shows the superior performance of prediction.

7. Data characteristics and managerial implications

In this section, we discuss the quality, characteristics, and managerial implications of the six cases of big data.

Data reliability. To begin with, the sources and quality of the six data sets are reliable. The first set of data about Twitter is organized by a respected university laboratory and an industrial partner. The cases of data about Facebook posts, Mashable blogs, and News in Google and Yahoo are widely downloaded by academics from a repository of big data (see the download statistics from the website of the UC Irvine Machine Learning Repository).

The four sets of data are successfully used by academic publications, see, e.g., Mayuri et al. (2015), Singh et al. (2015), Singh (2016), Fernandes et al. (2015), Moniz and Torgo (2018). The U.S. house data are collected by a national survey. The Bitcoin and news data are widely examined by academics and practitioners on the Kaggle website, which is the largest data community in the world with over one million users including many leading data experts.

Data quality. All data sets in our analysis do not have missing values, except the U.S. house data where we followed a similar procedure of Mullainathan and Spiess (2017) to clean the data including the removal of missing values, which represented a very tiny proportion.

Feature standardization. Furthermore, machine learning methods usually require users to standardize all features in a training data set into a standard normal distribution. This process avoids some features with large variances dominating the objective function of a machine learning method because the large variances prevent the model to learn from other features effectively. We use a preprocessing class *StandardScaler* in the *scikit-learn* package to complete the task of standardization, after which we can ignore the distribution form of data and focus on the prediction task.

Data characteristics. Next, we investigate the characteristics of the six data sets in terms of their cluster features. Naturally, some samples in a data set are located in an area where they are close to each other and are described as a cluster. Although it is not convenient to draw the cluster pattern for big data, we apply a clustering algorithm to fit the six sets of data. In this study, we use the k-means algorithm with all default parameter values provided by

Table 13. Cluster Characteristics of Six Data Sets

Case of Data Set	Data Cluster Characteristics: "Inertia" Attribute
Twitter Message	729,225,654,957.4763
Facebook Post	55,113,530,406,607,400
Mashable Blog	152,497,231,110,915.53
Google & Yahoo News	346,750.18708093156
U.S. House Market	7,316,325,268,523.303
Bitcoin and News	45.33088765908155

Notes. We use the k-means method with 8 clusters to fit the six data sets. The attribute of "inertia" is the sum of squared distances of samples to their closest cluster center. The distinctive values of "inertia" imply the diverse characteristics of the six data sets.

the *scikit-learn* package.

Particularly, the k-means method allocates the samples of each data set to 8 clusters. Based on the cluster allocation, the method can return the attribute of "inertia", which is the sum of squared distances of all samples in a data set to their closest cluster center. Therefore, we choose such attribute of "inertia" to represent the characteristics of the six sets of data. These are summarized in Table 13. The results show distinctive values of "inertia" which indicate the diverse characteristics of the six data sets. This demonstrates that our claim of "using partial samples provides superior performance" is robust to data characteristics which are crucial in big data validation.

Managerial implications. Finally, we discuss the managerial implications of our results relating to the problem objectives of the six cases of data under certain circumstances. Our results demonstrate that the sample size causes a tension and taking a partial sample, particularly around 20% of the full sample, usually brings superior prediction performance. For instance, let us consider the Twitter case as an example of the prediction of popularity. A manager at an organization or corporation can use the Twitter data to predict the popularity of a topic through 77 features that describe the evolution of these features through time t. For example, a feature represents the number of new authors interacting on a message's topic at time t. Our results show that when the manager uses the linear method of the elastic net with 20% of the full sample under a 75/25 train/test partition, a relatively high prediction performance can be achieved. The reason why the manager is advised to take this suggestion is that our scheme not only avoids unnecessary computing efforts to deal with the large full sample but also prevents the manager from making under-/over-estimation of the topic's popularity. Based on an accurate prediction of popularity, the manager would then be able to make appropriate decisions in his/her operating activities such as production and sale.

Similar discussions apply to popularity prediction with data from Facebook and Mashable. The results from the Facebook data show that a manager can obtain better prediction performance when applying linear methods or ensemble methods such as the random forest method while using 20% of the full sample under the train/test partitions with heavy weights on the training sets like 75/25 or 85/15. About the Mashable data, the manager can take 20% of the full sample when the SVM method is applied to a train/test partition of 55/45 with a relatively light weight on the training set.

In addition to popularity, a manager can use big data on news, government survey, and market prices to predict sentiment and price trends in markets. A manager or investor can use 20% of online news data that we examine to classify the optimistic or pessimistic categories of sentiment more accurately when employing ensemble methods or linear methods such as the ridge classifier while assigning more data allocations to the training set like the 85/15 train/test partition. Regarding the U.S house market, the investor could obtain gains in prediction when adopting the decision tree method and small partial samples. Similarly, the investor could predict the trends of Bitcoin prices more accurately based on partial samples from news data when the decision tree method or the ridge classifier is employed.

For example, Lazer et al. (2014) illustrate that the prediction from Google Flu Trends often leads to substantial errors due to the large size of data. Our method and finding shed light on the interesting example by providing suggestions to practitioners analyzing flu trends. The practitioners can run representative machine learning methods on an incremental percentage of the full sample both sequentially and randomly in order to identify an appropriate percentage of the full sample, likely around 20%, that would reduce the degree of overfitting and provide a high prediction performance.

Indeed, we reveal that on a number of occasions a better prediction performance can be achieved by using a proportion (approximately 20%) of the full sample rather than relying on a large sample. The relatively poor performance using a large sample is often due to the excessive, unnecessary and sometimes misleading information and noise that are embedded into the training process of a machine learning method. The search then incorporates all information resulting in a compromise to obtain a good overall result, which is not necessary the best.[10] We find that the result could be easily improved if less noisy data are introduced by only taking about 20% of the full sample. The underlying reason can be illustrated by a simple example. Suppose that there are 100 items of noisy data in a full sample with the size of 1000. The 100 noisy data disturb the machine learning method when the method tries to train a model to fit the full sample. On the contrary, taking 20% of the full sample probably carries only 20 items of noisy data whose disturbing effect on training is much weaker than the effect of 100 items of noisy data since the adverse effect of noisy data largely depends on the absolute quantity of noise rather than the relative level of noise in the sample. Hence, repeating the procedure of drawing 20% of the full sample randomly increases the chance of introducing less noise to training, which eventually improves the prediction performance.

In short, the six cases in our analysis cover different problems from various sectors and objectives. These include the predictions of the number of active discussion (NAD) to a Twitter topic, the number of comments received by a Facebook post, the number of shares of a blog on Mashable, sentiment implied by Google and Yahoo news, U.S. house prices, and Bitcoin prices. Our finding is therefore beneficial and worthwhile in guiding managerial decisions to be reached in a more appropriate way, which results in more robust solutions that are shown to be reliable in a variety of cases and data characteristics.

[10] A similar conclusion in the literature is that a model with a high level of complexity is liable to result in a poor prediction performance since it is distracted by noise or unnecessary details when it fits the complex model with data, see, e.g., Fenner (2019).

8. Conclusion and suggestions

The exponential growth of data from social media, news, and market provides users with diverse sources of big data. The application of machine learning algorithms to these big data provides new channels for organizations, individuals, and society to improve predictions and decisions (Wamba et al., 2015). Although we could embrace the exciting advantages of big data with enthusiasm, we need to be cautious about the tensions related to the application of machine learning.

The implicit premise of big data states that training more data could produce predictions with a higher accuracy but contemporary studies highlight that more data could also carry errors as well irrelevant information (Fan et al., 2014; Lazer et al., 2014; Hashem et al., 2015; Hák et al., 2016), which could unfortunately result in problems including overfitting issues (Hippert et al., 2005; Liu and Gillies, 2016; Silva et al., 2018; Jollans et al., 2019) and misleading biases (Corbett, 2018; Jollans et al., 2019). In our study, we investigate the tensions through six commonly used machine learning methods on real-world data sets purposely selected to represent a wide range of applications and diversity. We examine four representative regression methods for the data on Twitter, Facebook, and U.S. house surveys. We also apply four classification methods to the data on Mashable, Google and Yahoo News, and Bitcoin. Our numerical results reveal the tensions in big data that are caused by the different sensitivities of machine learning methods to several factors including the size of the training data set.

Recent developments in this particular research area related to problems and challenges arising in big data suggest some interesting ways such as dropping parts of the models (Ha et al., 2019) or around 20% of network nodes (Abdollahi et al., 2020), removing irrelevant data (Cai et al., 2018; Kanarachos et al., 2019) or irrelevant features (de Amorim, 2019). These research avenues are open questions that deserve a thorough investigation from both academics as well as practitioners. Our empirical results show that using approximately 20% of the full sample often provides performance gains compared with using the full sample. Our findings support the concerns raised by the above researchers. Meanwhile, it was also found that the relative advantages in the prediction performance of different methods vary with different data sets.

Moreover, our analysis under a variety of data sets and the train/test partitions provides suggestions to the application of machine learning to big data in social science. First, a user of machine learning can divide the full sample of a data set into a range of subsets with different percentages of the full sample. Second, he/she takes these subsets of data as the input of some representative machine learning methods. These subsets of data are further partitioned into training sets and testing sets under different partitions. Third, the user performs the methods on these training sets and then examines the prediction scores of these methods under different training sizes.

Fourth, the tension due to the availability of diverse methods can be resolved by comparing the performance of these methods. The representative methods can include several families of algorithms, e.g., general linear methods, tree methods, and ensemble-based methods. In addition, the user can run a comprehensive list of methods on a data set and check that the selected representative methods cover feasible ranges of accuracy variation and exclude those that are substantially inferior.

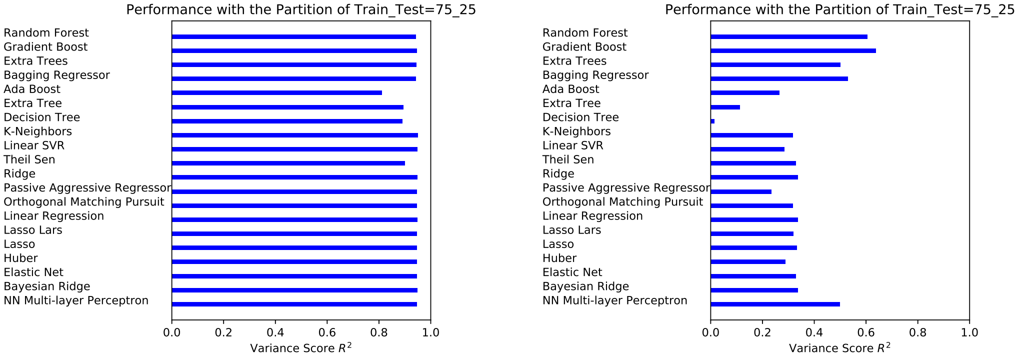
Fifth, the tension arising from the sensitivity of prediction accuracy to the training size can be resolved by firstly quantifying the deviation between the accuracy with a partial sample and the accuracy with a full sample. Then comparing the accuracy deviations for different sizes of samples reveals the training size that leads to high prediction performance.

In conclusion, though big data can obviously help organizations and individuals to make informed decisions (Blazquez and Domenech, 2018; Jun et al., 2018), it is also worth mentioning that one must be cautious about the tensions when applying machine learning methods to big data (Varian, 2014; Mullainathan and Spiess, 2017; Corbett, 2018). This is mainly because these methods are heuristic in nature and hence cannot guarantee optimality as many may not be aware of such a dilemma. However, these methods contribute to the best way forward to deal with such large data and hence need to be appreciated and deserve to be analyzed and studied even further. A systematic analysis of the prediction accuracy under a variety of training sizes will also indicate the way to resolve the tensions as empirically demonstrated in this study. We show that the tensions can be alleviated reasonably well if an appropriate fraction (or the use of a few fractions to be on the safe side) of the full sample is adopted instead of straightforwardly opting for the full size. In our experiments, around 20% of the sample shows to be reasonably promising. Our claim may also have some links with the well-established 20-80 Pareto Curve which one may like to explore further. Our finding, which is empirically proved on certain classes of instances, emphasizes that more is not always necessarily the best. In brief, we as academics and practitioners need to be aware of this hidden aspect whose effect ought not to be underestimated or taken lightly. More studies that explore this dedicate issue would, in our view, be a promising research area that is worth the challenge.

Appendix A: Performance of more machine learning methods

We discussed four regression methods and four classification methods in this paper. To provide a robust check, we use a number of machine learning methods under different families of algorithms to show that the methods discussed in the text generally cover feasible ranges of accuracy variation and exclude clearly worse methods for a data set. The regression methods include a neural network multi-layer perception regressor, ten general linear regressors, a stochastic vector regressor (SVR), a neighbor regressor, two tree-based regressors, and five ensemble-based regressors. The classification methods also consist of these families of algorithms for classification. In addition, we include two naive Bayes (NB) classifiers.

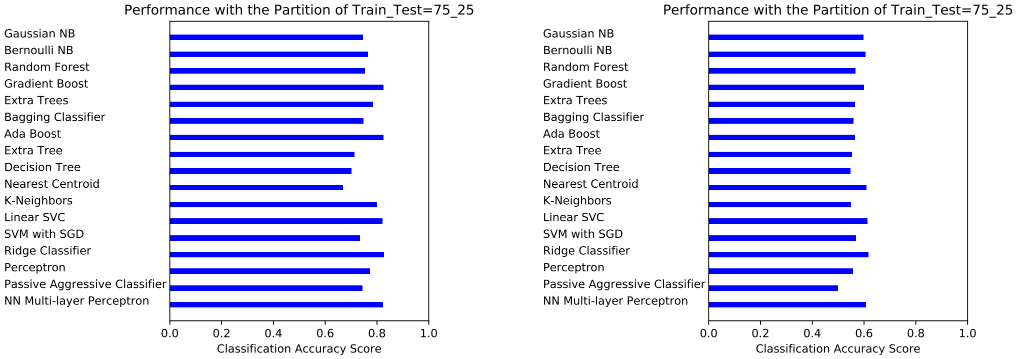
Figure 6 plots the regression variance scores using a variety of machine learning methods based on (a) Twitter data; (b) Facebook data while Figure 7 displays the classification accuracy scores using a variety of machine learning methods based on (a) Mashable data; (b) Google and Yahoo News data. In these experiments, we use the full sample of these data and split it into a 75% training set and a 25% testing set.



(a) Data of Twitter

(b) Data of Facebook

Figure 6. Performance of Regression Methods based on Twitter and Facebook Data.



(a) Data of Mashable Articles

(b) Data of Google and Yahoo News

Figure 7. Performance of Classification Methods based on Mashable and News Data.

Appendix B: Additional analysis on the consistency of sampling, parameters, and methods: Case of Bitcoin price prediction

In this appendix, we use the Bitcoin data to show that the machine learning methods that we examined consistently demonstrate a superior performance in prediction when partial samples are taken from the 20% to 80% of the full sample. These results are summarized in Table 14 to Table 17. The experiment for each column runs up to 100 times in total with random partial samples.

In addition, in Table 18 and Table 19, we also show that our conclusion is robust regarding the choice of the key parameter for both the decision tree method (DTM) and the random forest method (RFM). The total number of random runs for each column in the two tables is 100 times as well.

(a) Case of DTM- Here we investigate the parameter about the minimum number of samples that need to be kept in an internal node of a decision tree when the algorithm allocates the data into the tree structure.

(b) Case of RFM- In this scenario, we examine the number of the tree estimators that are built in an ensemble random forest estimator.

Furthermore, we point out that our findings are generalizable to most machine learning methods with a variety of complexities. The reason is that the methods in our experiment are taken from some families of methods and we show that each family of methods usually achieve similar performance, as shown in Figure 6 to Figure 7 in Appendix A. For example, the elastic net and the ridge classifier represent a large family of general linear methods while the random forest and the gradient boosting method are two representatives of ensemble methods. Meanwhile, since these methods cover various degrees of complexities, our conclusion is robust to the complexity of the problem as well.

In Table 20, we examine another five alternative methods to show that our conclusion can be generalized to other methods. Particularly, the passive aggressive classifier (resp. the bagging classifier) is another general linear (resp. ensemble) method. The K-Neighbors classifier is a widely-used k-nearest neighbors algorithm and the multi-layer perception is a typical neural network model. The last method is one of the naive Bayes classifiers. Almost all these alternative methods with 20% to 90% observations achieve more accurate predictions than those using 100% observations.

Table 14. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 20% Complete Sample 10 Times (Total 100 Random Runs)

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	4.85	4.14	8.03	5.50	4.73	5.30	4.97	8.29
std.	2.64	2.62	2.58	1.96	1.89	1.98	1.77	2.63
SVM with SGD	-0.32	1.31	4.18	2.40	1.16	2.92	0.51	2.32
std.	1.50	1.94	1.54	1.14	0.83	1.38	0.55	0.88
Decision Tree	4.67	2.57	4.12	1.11	1.17	2.37	3.44	3.69
std.	2.09	1.82	1.93	1.44	0.83	1.27	1.43	1.35
Random Forest	8.23	10.20	9.56	6.00	6.84	8.27	7.11	6.34
std.	3.35	3.89	3.23	2.71	2.62	2.75	2.28	2.38

Table 15. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 40% Complete Sample 10 Times

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	0.84	1.77	0.73	-0.17	-0.29	2.90	1.26	1.53
std.	1.00	1.01	0.70	0.57	0.35	0.93	0.54	0.60
SVM with SGD	1.41	3.16	3.99	2.11	1.66	2.70	2.43	2.28
std.	1.22	1.30	1.51	0.76	0.77	0.94	0.90	1.07
Decision Tree	1.79	2.74	3.52	3.18	1.55	4.78	3.90	3.87
std.	0.93	1.48	1.19	1.12	0.90	1.65	1.28	1.36
Random Forest	0.81	2.47	2.22	1.37	1.49	4.94	3.10	2.53
std.	1.04	1.76	0.89	0.79	0.93	1.53	1.06	0.86

Table 16. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 60% Complete Sample 10 Times

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	0.57	0.86	2.57	1.82	0.48	0.58	0.21	1.02
std.	0.81	0.78	0.90	0.75	0.48	0.47	0.28	0.42
SVM with SGD	2.32	2.19	2.50	1.04	0.77	0.55	0.67	0.43
std.	1.45	1.13	1.17	0.85	0.67	0.61	0.50	0.48
Decision Tree	1.45	0.17	1.77	1.75	0.54	0.67	1.25	1.51
std.	0.94	0.92	0.71	0.79	0.39	0.42	0.62	0.55
Random Forest	0.54	0.63	0.00	1.29	-0.50	-0.26	-0.47	0.21
std.	0.62	0.77	0.48	0.64	0.48	0.43	0.44	0.30

Table 17. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 80% Complete Sample 10 Times

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Ridge Classifier	-0.62	0.77	0.44	0.08	0.44	-0.47	0.77	-0.25
std.	0.85	0.37	0.62	0.29	0.28	0.34	0.40	0.21
SVM with SGD	0.75	0.62	0.19	0.33	0.83	1.55	0.34	0.93
std.	0.89	0.44	0.66	0.70	0.40	0.60	0.44	0.52
Decision Tree	-0.03	-0.52	-0.32	-0.57	-0.76	-1.09	-0.31	-0.43
std.	0.69	0.72	0.42	0.37	0.42	0.47	0.34	0.32
Random Forest	-0.77	-0.39	-0.64	-0.83	-1.27	-2.12	-0.99	-1.52
std.	0.64	0.56	0.52	0.36	0.59	0.71	0.42	0.55

Table 18. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 20% Complete Sample 10 Times

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Tree (2 samples/node)	3.20	3.51	1.00	3.20	-0.26	2.38	2.47	1.22
std.	2.00	1.45	0.85	1.57	0.70	0.90	0.93	1.00
Tree (4 samples/node)	2.76	3.51	1.24	1.80	-0.10	2.51	1.86	1.66
std.	1.99	1.52	0.94	1.47	1.13	1.02	0.84	0.90
Tree (6 samples/node)	5.40	6.11	3.14	5.00	2.37	5.64	6.50	5.54
std.	2.92	2.24	1.55	2.09	1.43	1.87	2.16	2.02
Tree (8 samples/node)	5.95	6.50	3.44	6.00	1.98	6.33	5.74	4.03
std.	2.88	2.55	1.72	2.29	0.96	2.08	2.00	1.56

Table 19. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 20% Complete Sample 10 Times

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Forest (3 estimators)	1.00	2.57	1.73	2.80	2.10	4.44	2.83	3.42
std.	2.01	1.78	1.43	1.79	1.46	1.47	1.03	1.21
Forest (5 estimators)	2.71	4.67	3.58	4.44	2.57	6.65	4.80	4.79
std.	2.82	2.03	1.62	1.92	1.68	2.24	1.62	1.62
Forest (7 estimators)	3.20	3.62	2.77	3.56	2.57	6.65	4.40	4.91
std.	2.76	1.80	1.62	1.88	1.51	2.18	1.53	1.63
Forest (9 estimators)	3.69	5.02	4.12	3.78	2.92	6.50	4.40	5.16
std.	2.71	2.00	1.77	1.93	1.39	2.14	1.48	1.68

Table 20. Deviation of Classification Accuracy (%) with Random Sampling 10 Times, Based on a Random 20% Complete Sample 3 Times using Five Alternative Methods (Total 30 Random Runs)

Fraction of Obs.	0.20	0.30	0.40	0.50	0.60	0.70	0.80	0.90
Passive Aggressive	15.56	33.02	23.70	42.22	26.90	29.31	31.85	22.89
std.	11.65	11.51	9.59	11.21	9.08	8.36	9.44	7.10
Bagging Classifier	5.78	13.23	12.84	10.37	14.39	7.41	8.77	13.11
std.	3.73	6.65	6.29	3.92	4.74	4.26	2.90	4.26
K-Neighbors	0.89	0.42	1.07	4.44	10.90	10.46	11.03	9.85
std.	5.86	4.79	4.76	2.90	5.00	5.50	4.32	3.44
NN Multi-layer Perceptron	-4.00	14.56	22.54	24.76	17.55	11.19	24.29	18.00
std.	8.02	5.16	7.60	8.08	6.95	5.36	7.29	5.65
Bernoulli Naive Bayes	8.00	2.86	2.96	-2.42	9.05	-5.00	4.07	-2.00
std.	5.55	4.69	4.32	1.91	3.33	3.61	2.37	1.79

References

Abdollahi, M., Khaleghi, T., Yang, K., 2020. An integrated feature learning approach using deep learning for travel time prediction. Expert Syst. Appl. 139, 1–11.

Bello-Orgaz, G., Jung, J. J., Camacho, D., 2016. Social big data: Recent achievements and new challenges. Information Fusion 28, 45–59.

Blazquez, D., Domenech, J., 2018. Big data sources and methods for social and economic analyses. Technol. Forecast. Soc. Chang. 130, 99–113.

Cai, J., Luo, J., Wang, S., Yang, S., 2018. Feature selection in machine learning: A new perspective. Neurocomputing 300, 70–79.

Corbett, C. J., 2018. How sustainable is big data? Production and Operations Management 27 (9), 1685–1695.

de Amorim, R., 2019. Unsupervised feature selection for large data sets. Pattern Recognit. Lett. 128, 183–189.

Economist, 2017. The world's most valuable resource is no longer oil, but data. https://www.economist.com/leaders/2017/05/06/the-worlds-most-valuable-resource-is-no-longer-oil-but-data.

Fan, J., Han, F., Liu, H., 2014. Challenges of big data analysis. Natl. Sci. Rev. 1 (2), 293–314.

Fenner, M., 2019. Machine learning with Python for everyone. Pearson Addison-Wesley.

Fernandes, K., Vinagre, P., Cortez, P., 2015. A proactive intelligent decision support system for predicting the popularity of online news. Proceedings of the 17th EPIA 2015 - Portuguese Conference on Artificial Intelligence, September, Coimbra, Portugal, 535–546.

Gan, L., Wang, H., Yang, Z., 2020. Machine learning solutions to challenges in finance: An application to the pricing of financial products. Technol. Forecast. Soc. Chang. 153, 119928.

George, G., Haas, M. R., Pentland, A., 2014. Big data and management. Acad. Manage. J. 57 (2), 321–326.

Goes, P. B., 2014. Editor's comments: Big data and IS research. MIS Quarterly 38 (3), iii–viii.

Ha, C., Tran, V.-D., Van, L. N., Than, K., 2019. Eliminating overfitting of probabilistic topic models on short and noisy text: The role of dropout. Int. J. Approximate Reasoning 112, 85–104.

Hák, T., Janoušková, S., Moldan, B., 2016. Sustainable development goals: A need for relevant indicators. Ecol. Indic. 60, 565–573.

Hashem, I. A. T., Yaqoob, I., Anuar, N. B., Mokhtar, S., Gani, A., Khan, S. U., 2015. The rise of "big data" on cloud computing: Review and open research issues. Information Systems 47, 98–115.

Hippert, H. S., Bunn, D. W., Souza, R. C., 2005. Large neural networks for electricity load forecasting: Are they overfitted? International Journal of Forecasting 21 (3), 425–434.

Hou, Y., Gao, P., Nicholson, B., 2018. Understanding organisational responses to regulative pressures in information security management: The case of a Chinese hospital. Technol. Forecast. Soc. Chang. 126, 64–75.

Iqbal, R., Doctor, F., More, B., Mahmud, S., Yousuf, U., 2020. Big data analytics: Computational intelligence techniques and application areas. Technol. Forecast. Soc. Chang. 153, 119253.

Jollans, L., Boyle, R., Artiges, E., Banaschewski, T., Desrivières, S., Grigis, A., Martinot, J.-L., Paus, T., Smolka, M. N., Walter, H., Schumann, G., Garavan, H., Whelan, R., 2019. Quantifying performance of machine learning methods for neuroimaging data. Neuroimage 199, 351–365.

Jun, S.-P., Yoo, H. S., Choi, S., 2018. Ten years of research change using Google Trends: From the perspective of big data utilizations and applications. Technol. Forecast. Soc. Chang. 130, 69–87.

Kanarachos, S., Mathew, J., Fitzpatrick, M. E., 2019. Instantaneous vehicle fuel consumption estimation using smartphones and recurrent neural networks. Expert Syst. Appl. 120, 436–447.

Kayser, V., Blind, K., 2017. Extending the knowledge base of foresight: The contribution of text mining. Technol. Forecast. Soc. Chang. 116, 208–215.

Lazer, D., Kennedy, R., King, G., Vespignani, A., 2014. The parable of Google Flu: Traps in big data analysis. Science 343 (6176), 1203–1205.

Liu, R., Gillies, D. F., 2016. Overfitting in linear feature extraction for classification of high-dimensional image data. Pattern Recognit. 53, 73–86.

Mayuri, M., Sneha, M., Kamatchi, P., 2015. Prediction of buzz in social-media using radial basis function neural networks. International Conference on Interdisciplinary Engineering and Sustainable Management Sciences 2015.

Miller, T. W., 2014. Modeling techniques in predictive analytics with Python and R: A guide to data science. Pearson.

Moniz, N., Torgo, L., 2018. Multi-source social feedback of online news feeds. https://arxiv.org/abs/1801.07055.

Mullainathan, S., Spiess, J., 2017. Machine learning: An applied econometric approach. J. Econ. Perspect. 31 (2), 87–106.

Olhede, S. C., Wolfe, P. J., 2018. The growing ubiquity of algorithms in society: Implications, impacts and innovations. Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences 376, 20170364.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research 12 (Oct), 2825–2830.

Reif, M., Shafait, F., Dengel, A., 2012. Meta-learning for evolutionary parameter optimization of classifiers. Machine learning 87 (3), 357–380.

Salhi, S., 2017. Heuristic search: The emerging science of problem solving. Springer.

Silva, S., Vanneschi, L., Cabral, A. I. R., Vasconcelos, M. J., 2018. A semi-supervised Genetic Programming method for dealing with noisy labels and hidden overfitting. Swarm Evol. Comput. 39, 323–338.

Singh, K., 2016. Facebook comment volume prediction. International Journal of Simulation: Systems, Science and Technologies 16 (5), 16.1–16.9.

Singh, K., Sandhu, R. K., Kumar, D., 2015. Comment volume prediction using neural networks and decision trees. IEEE UKSim-AMSS 17th International Conference on Computer Modelling and Simulation, UKSim2015, Cambridge, United Kingdom.

Thompson, A., 2017. All the news: 143,000 articles from 15 American publications. https://www.kaggle.com/snapcrack/all-the-news.

Varian, H. R., 2014. Big data: New tricks for econometrics. J. Econ. Perspect. 28 (2), 3–28.

Wamba, S. F., Akter, S., Edwards, A., Chopin, G., Gnanzou, D., 2015. How 'big data' can make big impact: Findings from a systematic review and a longitudinal case study. Int. J. Prod. Econ. 165, 234–246.

Wang, Y., Kung, L., Byrd, T. A., 2018. Big data analytics: Understanding its capabilities and potential benefits for healthcare organizations. Technol. Forecast. Soc. Chang. 126, 3–13.