Avrupa Bilim ve Teknoloji Dergisi
Sayı 20, S. 280-286, Aralık 2020

Araştırma Makalesi

www.ejosat.com ISSN:2148-2683

European Journal of Science and Technology
No. 20, pp. 280-286, December 2020

Research Article

The Role of Machine Learning in Productivity: A Case Study of Wine Quality Prediction

Ramazan Ünlü[1*]

[1] Gümüşhane Üniversitesi, İktisadive İdari Bilimler Fakültesi, Yönetim Bilişim Sistemleri Bölümü, Gümüşhane, Türkiye (ORCID: 0000-0002-1201-195X)

(First received 25 July 2020 and in final form 12 Ekim 2020)

(**DOI:** 10.31590/ejosat.773736)

ATIF/REFERENCE: Ünlü, R. (2020). The Role of Machine Learning in Productivity: A Case Study of Wine Quality Prediction. European Journal of Science and Technology, (20), 280-286.

Abstract

Machine learning has been used in many areas in recent years and has achieved quite successful results. Machine learning methods have been used from healthcare to driverless vehicles, it might also play a big role to increase productivity in the production sector. In this study, we have compared the performance of some machine learning strategies on an abnormally distributed dataset. Any machine learning methods can be easily applied to normally distributed data sets. However, it is necessary to alter the theoretical structure of the algorithm or data transformation process while a dataset is abnormally distributed. In this regard, three different methodologies are compared in this study. Initially, Support Vector Machines, which are often used in the literature, is used. Besides, Weighted Support Vector Machines, which is the revised version of the Support Vector Machines to produce successful results in abnormally distributed data sets. Finally, the Synthetic Minority Oversampling Technique (SMOTE) is applied, and the distribution of the dataset was artificially changed to normal distribution. Three techniques are compared in terms of sensitivity, specificity, precision, prevalence, F-1 score, and G-Mean evaluation criteria. Based on the results of the methods, Weighted Support Vector Machines produced the most successful results according to the chosen evaluation criteria.

Keywords: Support Vector Machines, Weighted Support Vector Machines, SMOTE

Verimlilikte Yapay Zeka'nın Rolü: Şarap Kalitesinin Tahminine Yönelik Bir Vaka Çalışması

Öz

Yapay zeka son yıllarda birçok alanda kullanılmaya başlanmış ve oldukça başarılı sonuçlar elde edilmiştir. Sağlık sektöründen sürücüsüz araçlara kadar birçok alanda kullanılan yapay zeka, üretim sektöründe de verimliliğin artırılması için sıklıkla kullanılmıştır. Bu çalışmada normal olarak dağılmamış bir veri setinde yapay zeka algoritmalarının kullanılmasına yönelik bir çerçeve çizilmeye çalışılmıştır. Normal dağılım gösteren veri setlerinde herhangi bir yapay zeka algoritması kolaylıkla uygulanabilirken normal dağılım göstermeyen veri setlerinde ya verinin kendisine farklı bir işlem uygulanması gerekir veya algoritmanın teorik yapısının revize edilmesi gerekmektedir. Bu açıdan bu çalışmada iki farklı yöntemde uygulanmıştır. İlk olarak literatürde sıklıkla kullanılan Destek Vektör Makinaları kullanılmıştır. Buna ek olarak Destek Vektör Makinalarının normal dağılmayan veri setlerinde başarılı sonuçlar vermesi için uyarlanmış şekli olan Ağırlıklandırılmış Destek Vektör Makineleri uygulanmıştır. Son olarak Sentetik Azınlık Aşırı Örnekleme Tekniği (SMOTE) tekniği uygulanmış ve kullanılan veri seti yapay olarak normal dağılıma yakınsanmıştır. Kullanılan üç teknikte duyarlılık, hassaslık, özgüllük, yaygınlık, F skor ve Geometrik Ortalama (G-Mean) değerlendirme kriterleri açısından karşılaştırılmıştır. Çalışma sonucuna göre Ağırlıklandırılmış Destek Vektör Makineleri kullanılan değerlendirme kriterlerine göre en başarılı sonuçları vermiştir.

Anahtar Kelimeler: Destek Vektör Makineleri, Ağırlıklandırılmış Destek Vektör Makineleri, SMOTE.

[*] Corresponding Author: Gümüşhane Üniversitesi, İktisadive İdari Bilimler Fakültesi,Yönetim Bilişim Sistemleri Bölümü, Gümüşhane, Türkiye, ORCID: https://orcid.org/0000-0002-1201-195X, ramazanunlu@gumushane.edu.tr

1. Introduction

Productivity can be defined as a measure of the efficiency of a person, machine, production systems, etc., in converting inputs into useful outputs. As different from a few decades ago, autonomous systems have become more popular and have been used in our age to increase the productivity of a system. As computers become capable of learning freely, reasoning, and determining the best course-of-action in real-time, they are started to be integrated into the real production systems to increase productivity. When it is said learning ability, the first things that come to the mind is artificial intelligence (AI) and machine learning (ML). Machine learning systems that take place under the umbrella of artificial intelligence provide splendid learning capabilities to computers. There available numerous machine learning approaches such as Multilayer Perceptrons, Support Vector Machines, Decision Trees, Clustering methods used in a broad spectrum of domains (Chen et al., 2017; Mondal et al., 2018; Segatori et al., 2018; Ünlü & Xanthopoulos, 2017, 2019).

Thanks to the advance of the learning algorithms over the last decade, machines might make a much more accurate prediction than a human in many applications. Based on the study of (Manyika, 2017), a machine can predict an image with a 26% error rate while human gives a 5% error in 2011. Thanks to proposed advanced methodologies, the error rate of autonomous machines decreased to 3% as of 2016. In the same study, they have claimed that automation will be a vital contributor to the productivity boost and autonomation of some sectors such as manufacturing, agriculture, transportation, and warehousing, etc., which will be inevitable in the future.

The ability of a machine is not limited to prediction from a dataset – that can eventually be thought of as solvable mathematical problems-, but they can beat a professional human in much more complex problems. The game field can be given as a sophisticated example, AlphaGo which is a collection of complex algorithms being able to learn how to play Go game beat Lee Sedol professional Go player in the Google Deep Mind challenge (Silver et al., 2016). These examples show us machine learning algorithms can be utilized in many different domains and might provide much more successful results than a human. From this perspective, machine learning techniques also can be implemented into the field in which productivity is an essential problem and one can increase the productivity of a system regardless of the complexity of the process.

In the literature, we can find various machine learning frameworks used to increase productivity. Xanthopoulos and Razzaghi developed a weighted support vector machine framework to identify machine failure (Xanthopoulos & Razzaghi, 2014). In that study, they worked on imbalanced datasets consist of different types of machine failures such as uptrend, downtrend, upshift, etc. They have proved that machine learning algorithms might work well even for a highly imbalanced dataset. This finding might help to increase productivity by surpassing human sense performance on identifying a rare failure occurrence. Another study of (Chalfin et al., 2016) demonstrated that machine learning techniques can help to improve the prediction of worker productivity. They tested the proposed framework in two important applications – police hiring and teacher tenure decisions.

In recent years, studies discussed machine condition monitoring and fault diagnosis due to potential advantages to be gained from decreased maintenance costs and increased productivity. Jack and Nandi, for example, used Support Vector Machines and Artificial Neural Networks to detect the fault of the roller bearing (Jack & Nandi, 2002). Sugumaran et al., (2007) have used a different method called decision tree to handle the same problem. Another important component affecting the productivity of a system in many industrial processes is the induction motor. Some studies have implemented machine learning methodologies to classify or predict the fault of an induction motor. To give examples, studies of (Fang, 2006; Poyhonen et al., 2002; Zhitong et al., 2003) have utilized the SVM method to handle the aforementioned problem.

Another interesting area of machine learning has touched is agriculture. Increasing agricultural productivity might be though as a global concern and producing enough and well quality products can help humans all over the world. For this, many machine learning-based approaches are used to increase agricultural productivity. Researchers have used different methodologies to provided rich recommendations and insights for farmers. Liakos et al., (2018) have reviewed machine learning related approaches used in agriculture in the study of. Based on his study, machine learning approaches are used for crop management, disease detection, weed detection crop quality, species recognition, livestock management, welfare and livestock production, water management, and soil management to increase agricultural productivity.

Based on some given examples from different fields, machine learning methodologies can give us good opportunities to improve the productivity of any application. For this reason, through our study, we have investigated several machine learning approaches and utilized them for a benchmark dataset to give a better insight into how to choose, apply, and evaluate a machine learning algorithm. We have tried to set a machine learning strategy to classify wines based on their qualities. This process is usually made by an expert. The main problem is the rate of undesired wines can be too low and a human can fail to detect it. So, we have tried to find the answer to the question of how well a machine learning system can be successful in terms of predicting the class of a wine which is under the desired quality level.

The following parts of the chapter are organized as follows. In section 2, we have explained the theory of the chosen machine learning methodologies. In section 3, we explained how we set up our experiment and give the results of the algorithms for the used dataset. Finally, we have discussed and concluded our study in terms of the context of productivity in section 4.

2. Material and Method

In our study, we have focused on an imbalanced dataset which is commonly seen in the operation of the production system (i.e. machine failure). Being able to handle with the imbalanced dataset can help to increase the productivity of the system. With this goal, we have investigated several strategies related to machine learning concept. The main algorithm chosen to utilize is Support Vector Machines which is a commonly used machine learning method in various applications. In addition to using baseline model SVM, we have applied weighting data samples and producing synthetic samples strategies. In what follows, we have explained the details of each strategy and discussed the main motivation behind them.

2.1. Data Preparation

Used dataset throughout the experiment is collected by (Cortez et al., 2009). This dataset is about the red and white wine quality and collected from May/2004 to February/2007 (Cortez et al., 2009). We chose the red wine quality consisting of 1599 samples with 11 different attributes and 6 different quality labels ranging from 0 to 10. Each wine sample was evaluated by a minimum of three sensory assessors by using blind tastes, and they are given a score ranging from 0 (very bad) to 10 (excellent). The final quality score is given by the median of these evaluations. To create a binary imbalanced dataset, we picked the wine samples that are labeled as 4 and 5. There are total of 734 samples of which 681 is quliality 5 and 53 is quality 4. The following Figure 1 illustrates the volume of chosen samples.

Figure 1. Frequency of Red Wine Samples

2.2. Support Vector Machines (SVMs)

The support vector machine is a commonly used ML method in literature and is considered one of the most influential algorithms. It is first proposed by (Boser et al., 1992). SVM has some theoretical advantages over some other ML algorithms such as the absence of local minima which might be a crucial problem for some methods such as Artificial Neural Networks. To mathematically formulate it assume that we have a training dataset {(x_1,y_2),….,(x_l,y_l)}, where each x_i ∈R^n and y_i ∈{-1,+1} . The main idea of the SVM is separating samples from different classes by finding a hyperplane whose distance is maximum concerning the data points of each class. The hyperplane can be defined by the parameters w and b. These parameters can be calculated by solving the following convex optimization problem 1 (see Equations 1a and 1b).

$$\min \frac{1}{2}\|w\| \tag{1a}$$

$$s.t. \quad y_i(w\phi(x)+b) \geq 1 \quad i=1,2,\dots l \tag{1b}$$

where ϕ denotes kernel function which is a non-linear mapping from R^n to R^m where m≥n. This problem is as known as hard margin SVM and if the classification problem is not separable, it is then infeasible. In this case, slack variables ξ_i,i=1,2,…l is added to the objective function, and the problem is transformed into the soft margin SVM. Soft margin SVM can be formulated as in optimization problem 2 (see equations 2a, 2b, and 2c).

$$\min \frac{1}{2}\|w\| + C\sum_{\{i=1\}}^{l} \xi_i \tag{2a}$$

$$s.t. \quad y_i(w^T\phi(x_i)+b) \leq 1-\xi_i \quad i=1,2,\dots,l \tag{2b}$$

$$\xi_i \geq 0, i=1,2,\dots,l \tag{2c}$$

where C refers to the magnitude of penalization. For faster and more stable convergence given optimization problem can be operated on the Lagrangian dual problem. The Lagrangian dual problem will be as shown in problem 3 (see Equations 3a, 3b, and 3c)

$$max \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{3a}$$

$$s.t. \quad \sum_{j=1}^{l}\alpha_i y_i = 0 \qquad i=1,2,\dots.l \tag{3b}$$

$$0 \leq \alpha_i \leq C \quad i=1,2,\dots l \tag{3c}$$

where $K(x_i,x_j)=\phi(x_i)^T\phi(x_j)$ is the kernel function that measures the similarity between two arbitrary points.

2.3. Weighted Support Vector Machines (WSVMs)

As can be seen in Equation 4, regular SVM assigns equal weight to data classes regardless of the size of them. This strategy can harm the overall performance of the algorithm (i.e. might cause overfitting). To handle with an imbalanced dataset, assigning different weights associated with the positive class size (C+) and negative class size (C-) are proposed by (Veropoulos et al., 1999). The optimization problem is revised as in Equations 4a, 4b, and 4c.

$$\min \frac{1}{2}\|w\| + C^+\sum_{\{i|y_i=+1\}}^{n_+}\xi_i + C^-\sum_{\{j|y_j=-1\}}^{n_-}\xi_i \tag{4a}$$

$$s.t. \quad y_i(w^T\phi(x_i)+b) \leq 1-\xi_i \quad i=1,2,\dots,l \tag{4b}$$

$$\xi_i \geq 0, i=1,2,\dots,l \tag{4c}$$

The problem can be transformed into the Lagrangian dual with the Kuhn–Tucker conditions. The Lagrangian dual is given by Equations 5a, 5b, 5c, and 5d.

$$max \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j K(x_i,x_j) \tag{5a}$$

$$s.t. \quad \sum_{j=1}^{l}\alpha_i y_i = 0 \qquad i=1,2,\dots.l \tag{5b}$$

$$0 \leq \alpha_i \leq C^+ \quad if\ y_i=+1\ and\ i=1,2,\dots l \tag{5c}$$

$$0 \leq \alpha_i \leq C^- \quad if\ y_i=-1\ and\ i=1,2,\dots l \tag{5d}$$

Thus, the main motivation behind assigning different weights to each class is forcing SVM to give more attention to minority class samples. C^+ and C^- is calculated as being inversely proportional to frequencies in the input data as shown in Equations 6a and 6b.

$$C^+ = n/(number\ of\ classes\ x\ n^+) \quad (6a)$$
$$C^- = n/(number\ of\ classes\ x\ n^-) \quad (6b)$$

Where n^+ and n^- represent the size of positive and negative classes respectively and n is the total number of samples.

2.4. Synthetic Minority Over-Sampling Technique (SMOTE)

Smote proposed by (Chawla et al., 2002) is another methodology to handle with the imbalanced datasets. The motivation behind it is to create synthetic data samples without replacement. The minority class is oversampled by taking each sample from the minority class and introducing the synthetic examples along the line segment of all the k nearest neighbors. For example, if one needs over-sampling 300%, only three of k nearest neighbors are chosen and a synthetic sample is created in the direction of each. Those samples are created in a way that taking the difference between the picked sample and (feature vector) and its nearest neighbor. Then, multiply this difference by a random number between 0 and 1, and add it to the feature vector picked at the beginning. This yields a selection of a random point on the line segment between the feature vector and its nearest neighbor.

2.5 Evaluation of the Methods

There available various evaluation metrics used for classification problems. One, maybe the most well-known- is the accuracy rate. It can be simply defined as the rate of correctly classified samples. This metric is powerful in the case of the balanced classification problem. However, accuracy might be misleading if the dataset is imbalanced. For example, assume we are given a dataset consisting of 99 positive samples and 1 negative sample. Predicting all samples as positive will yield %99 accuracy. Despite such a high accuracy rate, it cannot be concluded the model is successful because of being a failure of predicting rare cases.

Instead of using accuracy metrics, we can look at some other such as sensitivity, precision, specificity, F score, etc. To understand the concept of these terms, we need to first look at the confusion matrix. The confusion matrix is a table that is often used to describe the performance of the classification model on a test set in which ground true labels are known. The following Figure illustrates a confusion matrix for the binary classification problem. There are 4 different main arguments in a confusion matrix i) True Positive (TP): Cases in which the model predicts the class of samples as 1 and they are actually in a class of 1 ii) True Negative (TN): Cases in which the model predicts the class of samples as 0 and they are actually in a class of 0 iii) False Positive (FP): Cases in which the model predicts the class of samples as 1, but they are in the class of 0 iv) False Negative (FN): Cases in which the model predicts the class of samples as 0, but they are in the class of 1.

	Predicted 0	Predicted 1
Actual 0	True Negative (TN)	False Positive (FP)
Actual 1	False Negative (FN)	True Positive (TP)

Figure 2. Confusion Matrix

Based on these parameters, different metrics can be calculated. The following Table 1 shows the used evaluation metrics through our experiment.

Table 1. Formulations of the evaluation metrics

Evaluation Metrics	Formulations
Sensitivity/Recall	TP/Actual Positive
Specificity	TN/Actual Negative
Precision	TP/Predicted Positive
Prevalence	Actual Positive/Total number of samples
F1 score	2 x (Precision x Recall)/(Precision Recall)
G-Mean	$\sqrt{Sensitivity\ x\ Specificity}$

3. Results and Discussion

In this section, we have given the results of all methods. A well-known machine learning library Scikit-learn version of 0.19.2 running on the Python version of 3.6 is used for the experiment. Before utilizing any of the methods, the attributes of the dataset are normalized as being between 0 and 1. Also, as described above, three main strategies are applied which are SVM, WSM, and SVM with SMOTE. For the SMOTE process, we have oversampled minority class with a rate of 300% and called it SVM-SMOTE.

In addition to the mentioned pre-process of the data, we have applied parameter optimization by grid search method to get the optimum performance from the chosen methods. The performance of the methods is validated by the 5-fold cross-validation process. At this point, we can start by investigating our baseline model SVM. The following Table 2 shows the optimum parameters for the SVM model.

Table 2. Optimized parameters of the SVM model

SVM Parameters	Optimized values
C	1
Gamma	0.3
Kernel	Rbf

As we mentioned above, all classifier is run with 5-fold cross-validation to ensure the validity of the results. Using the optimized parameters given in Table 2, SVM produces the following results shown in Figure 3.

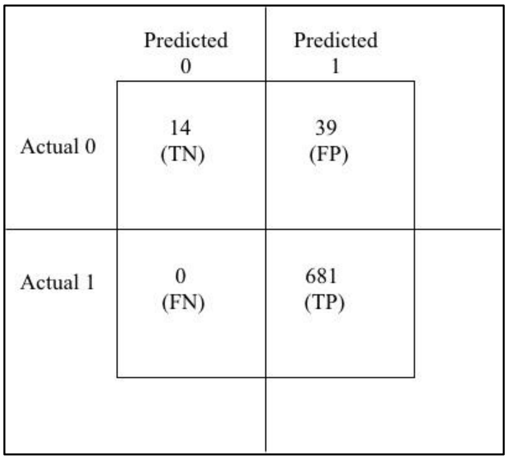

Figure 3. Confusion Matrix of SVM

Based on the confusion matrix given in Figure 3, our baseline model SVM gives high performance on predicting the examples in a positive class. However, we cannot say the same thing for the data samples in the negative class. Out of 53 samples from the negative class, it correctly predicted only 14 samples. In other words, specificity in the negative class is only 14/53=0.265 We can also infer from the confusion matrix misclassification rate is 39/734=0.5313. The following Table 3 gives the results of all calculated metrics which are determined based on the equations given in Table 1.

Table 3. The performance of SVM in terms of chosen evaluation metrics

Evaluation Metrics	Values
Sensitivity/Recall	1
Specificity	0.265
Precision	0.945
Prevalence	0.929
F1 score	0.971
G-Mean	0.514

According to the given results, we can conclude that SVM needs to be leveraged to make more trustworthy predictions. To do this, we have first implemented WSVM. The number of data samples in positive and negative of the classes (n^+ and n^-) are 53 and 681, respectively. Thus, the weight will be assigned to each sample in negative and positive class should be 781/(2*53)=7.368 and 781/(2*681)=0574. The following Table 4 shows the optimum parameters for the WSVM classifier. One needs to note that the C parameter in the optimization problem will be set as $class - weight_i \, x \, C$ for the class of i.

Table 4. The performance of WSVM in terms of chosen evaluation metrics

WSVM Parameters	Optimized values
C	0.2
Gamma	2
Kernel	Rbf
C+	0.574
C-	7.368

With the optimized parameters confusion matrix of the WSVM is shown in Figure 4.

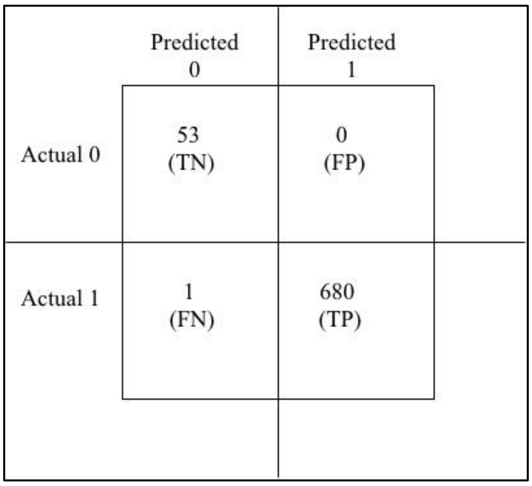

Figure 4. Confusion Matrix of WSVM

Based on the confusion matrix, WSVM inevitably produced much better results than SVM. It not only correctly predicts samples in positive class but shows high performance for samples in negative class as well. Out of 53 negative samples, it assigns all samples to the correct class. In other words, the specificity of the WSVM is 53/53. Without any doubt, it outperforms regular SVM in terms of the majority of the chosen evaluation metrics. The following Table 5 represents the performance of WSVM for all chosen evaluation metrics.

Table 5. The performance of WSVM in terms of chosen evaluation metrics

Evaluation Metrics	Values
Sensitivity/Recall	0.998
Specificity	1
Precision	1
Prevalence	0.929
F1 score	1
G-Mean	1

Based on the values shown in Table 5 WSVM is an excellent candidate to be the best classifier to given imbalanced data. The last method we have implemented is creating some synthetic points in the negative class by using the SMOTE method. As we mentioned above, we have oversampled the minority class with a rate of 300%. Meaning that we have created an extra 159 synthetics point in the minority class to make the dataset more balanced. The total number of samples now became 893. After doing that, we have again optimized the parameters as shown in Table 6.

Table 6. Optimized parameters of the SVM-SMOTE model

SVM Parameters	Optimized values
C	3
Gamma	0.5
Kernel	Rbf

The confusion matrix of the SVM-SMOTE model is shown in Figure 5. Based on the confusion matrix. We can say that the SVM-SMOTE model outperforms the regular SVM but cannot beat the WSVM. Its specificity rate is 208/212=0.98. Thus, it might be a highly effective classifier strategy for imbalanced dataset. The performance of SVM-SMOTE for all chosen evaluation metrics is represented in Table 7.

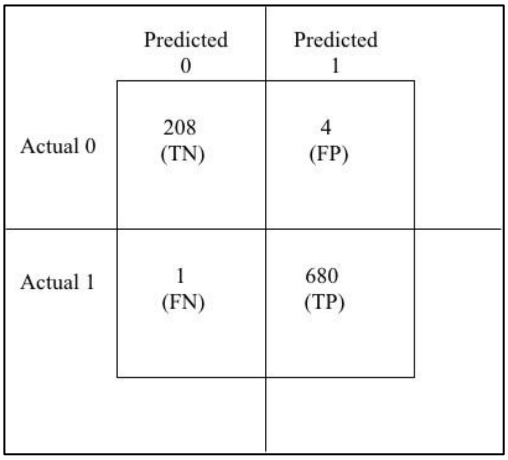

Figure 5. Confusion Matrix of WSVM

Table 7. Optimized parameters of the SVM-SMOTE model

Evaluation Metrics	The values
Sensitivity/Recall	0.998
Specificity	0.981
Precision	0.994
Prevalence	0.762
F1 score	0.995
G-Mean	0.989

To wrap up all results, we can say that WSM outperforms other classifiers, regular SVM and SVM-SMOTE. In terms of handling the imbalanced dataset, SVM does not perform well enough. Despite its predictions is good for the samples in the majority class, it fails to predict samples in the minority class. On the other hand, the SVM-SMOTE model is better than regular SVM. As it is expected, transforming the dataset as being more balanced makes it more separable. By doing this, the SVM-SMOTE model leveraged the performance of SVM. WSM, on the other hand, outperforms all the chosen method in terms of the evaluation metrics. It shows outstanding performance for predicting data samples in the minority class in addition to samples in the majority class.

4. Conclusions and Recommendations

Increasing productivity is one of the most important goals of the many systems (i.e. production). In our studied case, we have investigated the role of machine learning methods in terms of identifying a product that falls under the predetermined quality level. This kind of process is usually made by an expert in those specific products. Thus, the quality control process highly depends on the human sense. However, one of the biggest problems is that the frequency of undesired cases is likely too low during a production system. For example, a machine failure rarely occurs during the whole process, a product does not often fall a certain quality limit. This fact might make the identification of desired and undesired products by a human very hard. From this perspective, we have focused on how a machine learning algorithm can be used to successfully find rare cases. For this purpose, we have compared three methodologies called SVM, WSVM, and SMOTE. Based on our findings, an appropriate strategy can produce outstanding prediction results. The following Table 8 represents the comparison of methods in terms of the accuracy rate in the negative (rare cases) and positive class (i.e. specificity and sensitivity).

Table 8. The comparison of all methods

Evaluation Metrics	SVM	WSVM	SVM-SMOTE
Specificity	0.265	1	0.981
Sensitivity/Recall	1	0.998	0.998

Among the chosen methods, WSVM produces a very high classification rate in both negative and positive classes. Almost every single sample is correctly classified. Even though the SVM-SMOTE method produced inevitably good results, it did not outperform the WSVM methodology. However, we cannot the same thing for the SVM method. While it accurately predicts the samples in positive class, it fails to classify samples in the negative class. Hence, we can list the following arguments to wrap up our study

- The best method, in this case, WSVM, produced a 0.998% accuracy rate. This is an excellent classification rate, especially for the imbalanced dataset. It also might be much better than the accuracy rate of a human sense. That is why we can conclude that a well-designed machine learning strategy can help to investigate undesired products, and so it might help to companies/systems to increase the total productivity.
- This study has focused on the binary classification problem (desired and undesired products). However, it can be easily extended to a multiclass classification

problem. This might help to classify a range of products with different quality standards.

Explained strategies can be used not only to find undesired products but predicting some important events which might affect the productivity of a system. For example, correctly predicting a machine failure before it occurred might help to take precautions. By doing this, the overall productivity of a system can be increased by avoiding long time system disruption. For future research, we aim to extend this framework to a multiclass classification problem. Besides, a state-of-art data creation method can help better to obscure imbalancedness of a dataset.

References

Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. Proceedings of the Fifth Annual Workshop on Computational Learning Theory, 144–152.

Chalfin, A., Danieli, O., Hillis, A., Jelveh, Z., Luca, M., Ludwig, J., & Mullainathan, S. (2016). Productivity and selection of human capital with machine learning. American Economic Review, 106(5), 124–127.

Chawla, N. V, Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321–357.

Chen, Q., Xu, J., & Koltun, V. (2017). Fast image processing with fully-convolutional networks. Proceedings of the IEEE International Conference on Computer Vision, 2497–2506.

Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. Decision Support Systems, 47(4), 547–553.

Fang, R. (2006). Induction machine rotor diagnosis using support vector machines and rough set. International Conference on Intelligent Computing, 631–636.

Jack, L. B., & Nandi, A. K. (2002). Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. Mechanical Systems and Signal Processing, 16(2–3), 373–390.

Liakos, K., Busato, P., Moshou, D., Pearson, S., & Bochtis, D. (2018). Machine learning in agriculture: A review. Sensors, 18(8), 2674.

Manyika, J. (2017). A FUTURE THAT WORKS: AI, AUTOMATION, EMPLOYMENT, AND PRODUCTIVITY.

Mondal, A., Ghosh, A., & Ghosh, S. (2018). Scaled and oriented object tracking using ensemble of multilayer perceptrons. Applied Soft Computing, 73, 1081–1094.

Poyhonen, S., Negrea, M., Arkkio, A., Hyotyniemi, H., & Koivo, H. (2002). Fault diagnostics of an electrical machine with multiple support vector classifiers. Intelligent Control, 2002. Proceedings of the 2002 IEEE International Symposium On, 373–378.

Segatori, A., Marcelloni, F., & Pedrycz, W. (2018). On distributed fuzzy decision trees for big data. IEEE Transactions on Fuzzy Systems, 26(1), 174–192.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., & Lanctot, M. (2016). Mastering the game of Go with deep neural networks and tree search. Nature, 529(7587), 484.

Sugumaran, V., Muralidharan, V., & Ramachandran, K. I. (2007). Feature selection using decision tree and classification through proximal support vector machine for fault diagnostics of roller bearing. Mechanical Systems and Signal Processing, 21(2), 930–942.

Ünlü, R., & Xanthopoulos, P. (2017). A weighted framework for unsupervised ensemble learning based on internal quality measures. Annals of Operations Research, 1–19.

Ünlü, R., & Xanthopoulos, P. (2019). Estimating the number of clusters in a dataset via consensus clustering. Expert Systems with Applications.

Veropoulos, K., Campbell, C., & Cristianini, N. (1999). Controlling the sensitivity of support vector machines. Proceedings of the International Joint Conference on AI, 55, 60.

Xanthopoulos, P., & Razzaghi, T. (2014). A weighted support vector machine method for control chart pattern recognition. Computers & Industrial Engineering, 70, 134–149.

Zhitong, C., Jiazhong, F., Hongpingn, C., Guoguang, H., & Ritchie, E. (2003). Support vector machine used to diagnose the fault of rotor broken bars of induction motors. Electrical Machines and Systems, 2003. ICEMS 2003. Sixth International Conference On, 2, 891–894.