Predicting Car Sale Time with Data Analytics and Machine Learning

Hamid Ahaggach[1(✉)], Lylia Abrouk[1(✉)], Sebti Foufou[1(✉)], and Eric Lebon[2(✉)]

[1] LIB Laboratory, University of Burgundy, Dijon, France
{Hamid.ahaggach,lylia.abrouk,sfoufou}@u-bourgogne.fr
[2] Syartec, Aix en Provence, France
elebon@syartec.com

Abstract. There is no doubt that marketing is an important step in Product Lifecycle Management (PLM) and obviously decreasing time-to-market is crucial to reduce storage costs and increase profit. This paper aims to improve marketing strategies in the automotive field for car dealers and car selling supply chain. Due to the cost of new cars and the high risk of car value depreciation it becomes necessary for car dealers to know which type of cars can be sold faster than others, this will allow dealers to adapt their marketing strategies and satisfy the need of their customers. We propose to use data analysis and machine learning algorithms to address this problem and create models to help these companies in their decision-making processes. In our experiments, we used sale data from two big dealers of multi-maker cars. The dataset contains the sale history of around 73200 cars over a period of 8 years. We compared the different machine-learning algorithms and got promising results classifying cars into different predicted sale time ranges.

Keywords: Product Lifecycle Management · Data analysis · Machine learning

1 Introduction

Product Lifecycle Management (PLM) is an important strategy that helps companies in the representation, storage, and processing of product information throughout its lifecycle phases [1]. At every stage of the lifecycle, PLM solutions ensure integrated processing of all relevant information. Although the term PLM is much more used in the manufacturing sector than any other sector, PLM is expanding to cover more and more broader applications such as software industry and marketing strategies.

PLM organizes and integrates data to provide a detailed view of each product manufactured and how it is received in the marketplace, maximizing efficiency in the following areas [2]:

- Design and manufacturing integration: A company's production process can use a range of software applications for design and manufacturing. With PLM,

Published by Springer Nature Switzerland AG 2023
F. Noël et al. (Eds.): PLM 2022, IFIP AICT 667, pp. 399–409, 2023.
https://doi.org/10.1007/978-3-031-25182-5_39

the entire production process can be optimized in real time. Without PLM, valuable data may not be shared across all these systems and people.
- Virtual environments that support global operations: PLM provides a central repository for product data management (PDM) that integrates the entire global process from concept to customer.
- Accessible data: PLM makes all product information available to each department, improving production efficiency and allowing for a closed-loop for all teams involved.
- Product commercialization: PLM assures products are ready for global rollout, with effective, reliable data, document management and process governance. Unified data and collaborative workflows across the organization keep things running smoothly and allow teams to respond quickly when challenges arise.

In this paper, we consider the car commercialization aspect of PLM to help solving the inventory problems of car dealers. Indeed, car dealerships buy cars from the manufacturers and sell them to their customers to make profit. Obviously, dealers cannot just send the unsold cars back to the manufacturer. Keeping unsold cars in the parking lots for more than six months is extremely costly and may even threaten the financial prosperity of these companies. Therefore, they will have to find a way to get rid of these cars and get prepared to receive newer models, but this cannot be done without losing a lot of money. This paper proposes to use data analytics and machine learning to predict the time required to sell car models. Car properties and sales history are taken into consideration to compute ML models capable of making the correct prediction in most of the cases. The contributions of this paper can be summarized as follows: (i) using data analytics to find car properties with the highest influence on car sales, (ii) build several ML algorithms to predict car selling time, (iii) conduct experiments to test these algorithms, compare and discuss their results.

The rest of the paper is organized as follows. Section 2 presents the state of the art of using Machine-learning in PLM. Section 3 describes our method to improve marketing strategies in the automotive field for car dealers. Section 4 presents and discusses the experimental results obtained with the proposed Machine Learning based algorithms. Section 5 concludes the paper and gives few perspectives to extend and improve this work.

2 PLM and Machine Learning

As Big Data becomes more prevalent, Machine Learning (ML) is opening up new opportunities for data processing to support decision making in all areas of manufacturing, from customer engagement to design and supply chain to product lifecycle management. The combination of Big Data and advanced, low-cost computing systems has made machine learning viable in many real world work applications, e.g. cultural heritage [23], leading to positive changes in the product development lifecycle [3]. Today, companies are entering a new era of digital transformation in product lifecycle management (PLM) where ML is one of the enablers of forth Industrial Revolution (usually called *Industry 4.0*), which is almost twice as likely to be used as any other tool [3]. Currently, ML models used to enhance the PLM power in offering additional insights into the product lifecycle [3].

2.1 Machine Learning in Product Design

ML algorithms are used to support product design at least in two major aspects: analyze market trends and assist designers to achieve a fast and personalized design processes. The conceptual design stage is increasingly seen as an essential step in product development and customization. Effective conceptual design is inseparable from proper market investigation, as it has a critical impact on market prospects, customer acceptance and product lifecycle, among the works realized in this context we mention [4,5]. Market analysis aims to identify target customers, recognize their requirements, and translate these requirements into product features. Compared to manual market analysis, market research based on data mining can discover the implicit associations of market data by integrating ML and analytic algorithms such as Support vector machine [6], Apriori [7], ARIMA [8].

2.2 Machine Learning in Product Manufacturing

ML improves the product manufacturing framework, including material procurement, resource configuration, production scheduling, machining, assembly, quality control, warehousing, logistics, etc. ML contributes to improving the manufacturing phase of products in two ways: optimizing the information flow within the manufacturing processes, and monitoring smart devices to execute repetitive tasks. In addition, the integration of ML and production systems enables significant advances in human-machine collaboration, defect prediction, intelligent decision-making, and other aspects, Among the works realized in this context we mention intelligent supplier selection decision [9,10], Human-robot collaborative manufacturing control [11,12].

2.3 Machine Learning in Product Services

The combined application of Deep Learning and recommendation algorithms (e.g., Deep Neural Networks (DNN) and collaborative filtering) can improve the personalized decision level of recommendations by analyzing user and product group information associated with demand [14]. Moreover, the time series network can extract preference features based on the user's historical purchase behavior and make recommendations by modeling the content of the product sale network and user profile [15]. Moreover, product services need technical guidance for product maintenance according to consumer requirements. Traditionally, it is difficult for customers to perceive gradual changes in the condition, quality, and performance of products. Therefore, unnoticed product deterioration affects the user experience and reduces the quality of customer service. That is why product status monitoring is an important part of product lifecycle information processing as it is used to evaluate equipment performance and prevent product failures [16,17].

3 Methodology

This section presents the proposed solution to help car dealers in their effort to car inventory problems. As stated in the introduction section dealerships cannot afford keeping unsold cars in their parking lots for an extended period, so effective solutions must be offered to find out which cars could be sold within a short period of time, and which cars could require longer sale time. This classification will allow dealerships to put better marketing strategies for example reducing the number of car models that are not easy to sell.

Time series could be used as a tool to predict which cars will be sold in the next months, but this method is not practical in our case because the sales information, in the datasets we are using, do not follow a precise pattern. For example, if we take car C that was sold once since the company was established, where $\{c_1, c_2, \ldots, c_t\}$ are the monthly sale values, then most of the values are equal to zero. Given the sale values $\{c_s, \ldots, c_e\}$ over a period $[s, e]$, where c_s and c_e are respectively the sale value of the start and the end of the period, if we train a network based on sequential models like *(LSTM, RNN ...)*, or based on a statistical analysis model like *ARIMA* to predict $Y(t+1) = F(c_s, \ldots, c_e)$, the sale value for the next period $t+1$, then we will get wrong results. Therefore, we propose to use car characteristics to predict selling time. So we have dataset of pairs $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ where $X = \{x_1, x_2, \ldots, x_n\}$ contains the characteristics of cars such as the color, the price, the power of the engine... and $Y = \{y_1, y_2, \ldots, y_t\}$ is the time taken to sell the vehicles (see Table 1). We need to find function $f(x_i) = y_i$ that will be able to predict the time needed to sell a car, this function can be any machine-learning algorithm. As shown in Fig. 1 the proposed system operates in three steps: (i) data processing, (ii) dimensionality reduction, and (iii) model training. In the next sections, we will describe the dataset, the machine learning models and highlight their similarities and their differences.

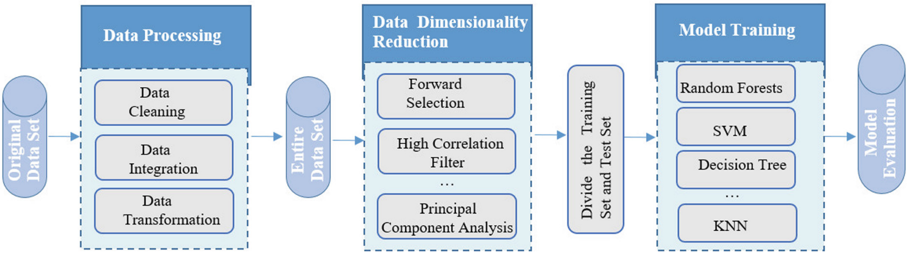

Fig. 1. The system architecture of the proposed solution

3.1 Dataset

The dataset used in this work was provided by two major multi-brand car dealerships in the European Union. The dataset covers the dealerships activities for a period of 8 years from Oct. 2013 to Nov. 2021. It contains more than 73200 data samples and 33 attributes for each data sample. A brief description of these attributes is given in Table 1.

Table 1. Attributes of car sales dataset.

Attributes	Description	Example
Label	Label of vehicle: 0: if selling the vehicle takes less than 3 months 1: if selling the vehicle takes between 3–6 months 2: if selling the vehicle takes between 6–8 months 3: if selling the vehicle takes between 8–12 months 4: if selling the vehicle takes more than 12 months	0
Entry_date	The vehicle entry date	2013-10-10
company_name	Name of company	Com1
Sale_date	The vehicle sale date	2013-12-02
Carrosserie	Vehicle bodywork	COUPE
C02	C02 emissions in g per km	185
Couleur_Vehic	Color of vehicle	GRIS F
Cylindree	The volume of gas that can be burned in the cylinders	1995
Date_1ere_Cir	Date of 1st entry into service	20040629
Depollution	Depollution device	OUI
Empat	Wheelbase of vehicle	273
Energie	Fuel type of vehicle	GAZOLE
Genre_V	Type of vehicle	VP
Immat	Registration of vehicle	BX368SQ
Marque	Marker of vehicle	BMW
Modele	Model of vehicle	SERIE 3
Nb_Cylind	Number of cylinder	4
Nb_Pl_Ass	Number of seats or payload	5
Nb_Portes	Number of doors	2
Nb_Soupapes	Number of valves	4
Nb_Vitesses	Number of speeds for manual gearboxes	5
Propulsion	Propulsion	ARRIERE
Puis_Ch	Real power in steam horsepower	150
Puis_Fisc	Fiscal power in fiscal horsepower	9
Tp_Boite_Vit	Gearbox type	B.V.A.
Turbo_Compr	Presence of turbo	TURBO
Version	Version of vehicle	320 CD
Code_Moteur	Motor code	204D4/M47TUD20
Cons_Urb	Urban consumption	9,7
Cons_Exurb	Extra-urban consumption	5,4
Cons_Mixte	Mixed consumption	6,9
Prix_Vehic	The selling price of the vehicle	28 800 €
C_code	If the vehicle is used or new	NV

3.2 Data Processing

This step is very important and sensitive because it directly affects the results of the model; in our case, we noticed missing data for several attributes. For example attributes such as *Couleur_Vehic* and *Cons_Exurb* are lucking about 5000 data, but luckily the missing data is not so important for all the attributes, e.g. There are less than 10 missing data for attributes *Tp_Boite_Vit*, *Energie* and *Code_Moteur*. Depending on the influence of different attributes on car sales and the amount of missing data, we treat the problem in different ways, for attributes with a high number of missing data and little impact on car sales, for example, *Color_Vehic*, we first group the data by *Label*, then fill each group's attribute by the mode value. For attributes with some missing data but having a significant impact on car sales, we design an accurate filling method. For anomalous data and missing data that cannot be filled, we opt for zero filling to minimize their impact on prediction accuracy. In addition, some data elements are anomalous due to possible recording errors and must be filtered out. We also noticed the presence of data of the same type but written in different formats for example *Sale_date* and *Entry_date* written respectively in *YYYY-MM-DD* and *MM/DD/YY* formats. A data integration step is therefore required to address these kinds of discrepancies and bring attributes of the same type into a unique format. In addition, non-numerical data cannot be used directly for prediction as the same information is rarely expressed in a unique way e.g. passenger front door, passenger side door, front right door, right front door are all used to refer to the same information, therefore, it is essential to transform the non-numeric data into numeric data in a way that best preserves the information and facilitates feature extraction.

3.3 Dimensionality Reduction

With such a large amount of data, the variety of attributes and their formats, we must carefully select the intrinsic features to achieve a high prediction accuracy while reducing computation costs. We also need to apply a dimensionality reduction method to allow visualizing the data in a reduced space. We distinguish two classes of dimensionality reduction methods: Methods of the first class keep only the most important features in a dataset and eliminate the rest; in this case, the features are not transformed. Backward elimination, Forward selection and Random forests are examples of this method. Method of the second type find a new combination of features, in this case, features are transformed, and the new set of features contains different values instead of the original values. We can divide this class further into linear and non-linear methods. The Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA) are examples of linear dimensionality reduction methods. t-distributed Stochastic Neighbor Embedding (t-SNE) is an example of non-linear dimensionality reduction methods, In Fig. 2, we display the data of new vehicles of the first company in 2D space with PCA.

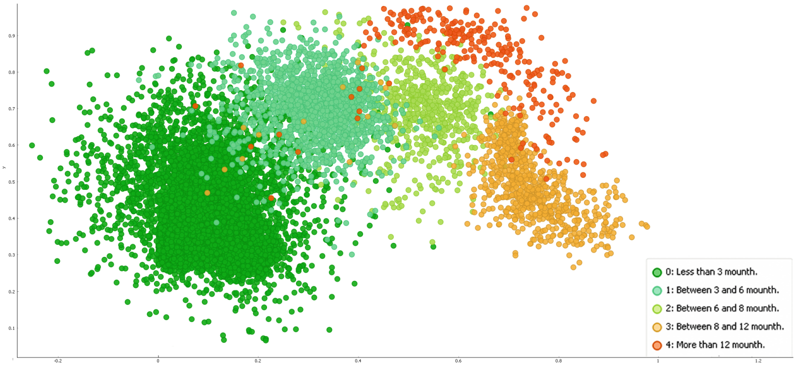

Fig. 2. Scatter plot of data of the new vehicles for the first company in 2D space after PCA dimensionality reduction.

3.4 Machine Learning Algorithms

To predict whether a car would, or would not, be sold in a predetermined period of time we use classification algorithms. To this end, the different classes of cars are identified, and a label is assigned to each car in the dataset to mark to which class it belongs. This type of learning is named "supervised" because we give our algorithm the data with their labels to learn, and once the model is learned, it will be able to predict the label of a car never seen before. Among few others, we have tried the below classification algorithms.

Support Vector Machines (SVM) is the best-known form of kernel methods inspired by Vladimir Vapnik's statistical theory of learning. SVM is a method of classification by supervised learning introduced by Vapnik in 1995 [18]. This method searches for the hyperplane that separates the positive data samples from the negative ones, ensuring that the margin between the closest positive and negative is maximal. This ensures a generalization of the principle because new examples may not be too similar to those used to find the hyperplane but may be located on one side or the other of the border. The interest in this method is the selection of support vectors, which represent the discriminant vectors by which the hyperplane is determined. The examples used during the search for the hyperplane are then no longer useful and only these support vectors are used to classify a new data sample, which can be considered as an advantage for this method.

Decision trees (DT) is a supervised learning technique that can be used for classification and regression problems, it is called a decision tree because, similar to a tree, it starts with the root node, which grows on other branches and builds a tree structure. In a decision tree, there are two type of nodes, which are the decision nodes and the leaf nodes. The decision nodes are used to make any decision and have several branches, while the leaf nodes are the output of these decisions and do not contain other branches. To build a tree and find out the attribute that should be selected for its initialization one can use the CART algorithm [19], which is based on the Gini index, or the ID3 or C4.5 algorithms, which are both based on the notion of entropy [20,21].

Random forests (RF) are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [22]. The generalization error for forests converges to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them.

K-Nearest Neighbors (KNN) is one of the simplest machine learning algorithms, based on the supervised learning technique. KNN stores all available data and classifies a new data point based on similarity. This means that when a new data point appears, it can be easily classified into a well-fitting category using the KNN algorithm.

4 Experimental Results

To test the above algorithms and compare their prediction results, we use a large-scale dataset provided by two car dealership companies covering their car sale activities for the period between Oct. 2013 and Nov. 2021. The dataset has 33 attributes and more than 73200 entries. The dataset of the first company contains 40700 among these cars there are 18800 new cars and 21900 used cars, and for the second company there are in total 32500 cars among which there are 18700 new cars and 14000 used cars. Our goal is to predict the time margin that a car will stay in stock before being sold, we will build two models for each company, one model for used cars and the other for new cars. The dataset is randomly split into two parts: training set (80% of the dataset), is used to train and test set (20% of the dataset) to evaluate our model, during the train we validate our training process using 10-Folds cross-validation. The accuracy and training-time are considered as a comparison criterion between algorithms on the test set. The accuracy in our case is defined as follows:

$$Accuracy = The\ number\ of\ well{-}classified\ cars/The\ total\ number\ of\ cars \tag{1}$$

The results of our experiments are reported in Table 2. To train our models we used a Dell OptiPlex-7070 computer with Intel(R) Core(TM) i7-9700 @ 3.00GHz 8-Core CPU and 16GB DDR4 RAM on Windows 10 Pro 64-bit.

Table 2. Results of the prediction on the datasets of the two companies.

Company	Vehicle type	Metrics	Models			
			KNN	*SVM*	*DT*	*RF*
1	VN	Accuracy	0.971	0.951	**0.990**	**0.990**
		Training time	**0.096 s**	8.431 s	0.996 s	0.140 s
	VO	Accuracy	0.849	0.854	0.814	**0.863**
		Training time	0.058 s	13.52 s	**0.057 s**	0.647 s
2	VN	Accuracy	0.967	0.944	0.987389	**0.994**
		Training time	**0.079 s**	20.76 s	0.140 s	1.145 s
	VO	Accuracy	0.845	0.862	0.802	**0.870**
		Training time	**0.066 s**	47.15 s	0.187 s	1.484 s

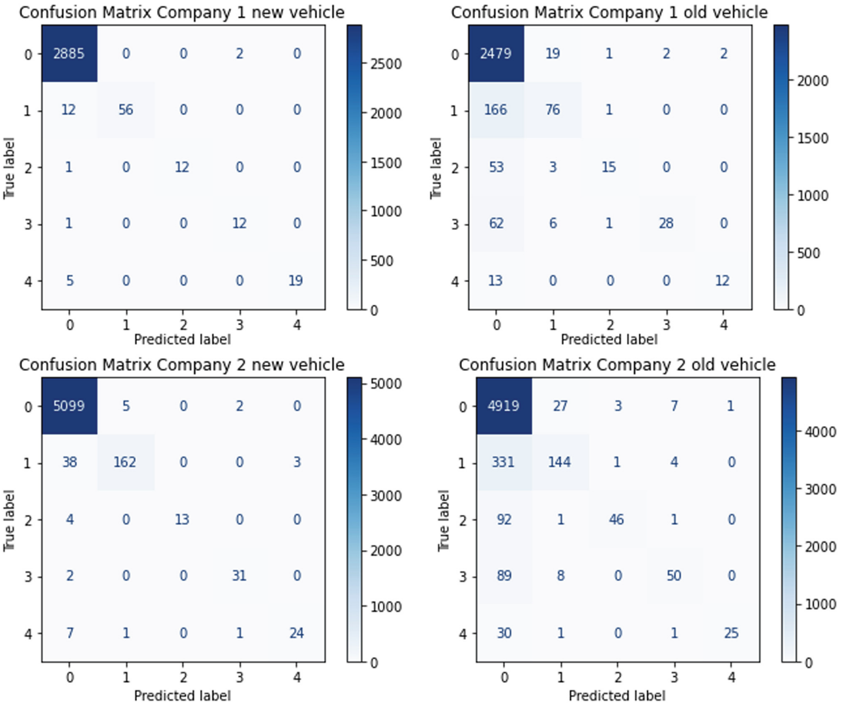

Fig. 3. Confusion matrix of the prediction on the two datasets

Table 2 shows that the random forest gives much better results in comparison with other models, and this is because RF is composed of several decision trees that collaborate with each other. In the case of the first company, both the DT and RF give the same accuracy score on a new vehicle because the data in this case are easy to be discriminated by the decision tree. We also note that KNN generally gives good results because it is based on data. SVM takes a lot of time to learn in comparison with other models because SVM tries to find a hyperplane that separates the margin between the nearest samples of each class, generally maximization problems take more time and depend on the performances of the machine used to train the model. In Fig. 3, the confusion matrix for each type of cars for the two companies with the model that we found the best score. One can see that the models are able to classify the cars in the correct category. There is a little confusion between class 0 and 1. That is because we can find two cars of the same type, one sold in three months and the other in three months and a day, for the model, they are two cars of two different classes. This kind of confusion are solvable with regression models: instead of predicting the sales time intervals we will predict the exact number of days to sell a car, but generally companies are not interested in predicting the exact day but the time interval of sales.

5 Conclusion and Perspectives

In this paper, we proposed to implement SVM, DT, KNN, and RF machine learning algorithms to predict the time required for dealers to sell cars. A large-scale car sales dataset provided by two multi-maker dealership companies has been pre-processed to complete missing data and identify the car characteristics that have the greatest impact on car sales. This sale time prediction gives companies better ideas about the commercialization of vehicles and hence help them putting the right marketing strategy to avoid buying cars that are not easy to sell. In future work, we intend to extend this work towards customer behavior analysis to build a recommendation system based on association rules, to target customers who can buy specific cars based on the profile of former customers.

References

1. What is product lifecycle management. https://www.sap.com/insights/what-is-product-lifecycle-management.html. Accessed 9 Dec 2021
2. Product lifecycle management. https://www.propelplm.com/articles/what-is-product-lifecycle-management. Accessed 2 Mar 2022
3. PLM and machine learning meet. https://www.aberdeen.com/featured/blog-when-plm-machine-learning-meet/. Accessed 29 Feb 2022
4. Hu, X., Hu, J., Peng, Y., Cao, Z.: Constrained functional knowledge modelling and clustering to support conceptual design. Proc. Inst. Mech. Eng. C J. Mech. Eng. Sci. **226**(5), 1326–1337 (2012)
5. Liu, X., Liu, H., Duan, H.: Particle swarm optimization based on dynamic niche technology with applications to conceptual design. Adv. Eng. Softw. **38**(10), 668–676 (2007)
6. Pal, R., Kupka, K., Aneja, A.P., Militky, J.: Business health characterization: a hybrid regression and support vector machine analysis. Expert Syst. Appl. **49**, 48–59 (2016)
7. Kumar, V.S., Renganathan, R., VijayaBanu, C., Ramya, I.: Consumer buying pattern analysis using apriori association rule. Int. J. Pure Appl. Math. **119**(7), 2341–2349 (2018)

8. Gurnani, M., Korke, Y., Shah, P., Udmale, S., Sambhe, V., Bhirud, S.: Forecasting of sales by using fusion of machine learning techniques. In: 2017 International Conference on Data Management, Analytics and Innovation (ICDMAI), pp. 93–101. IEEE (2017)
9. Keskin, G.A., İlhan, S., Özkan, C.: The Fuzzy ART algorithm: a categorization method for supplier evaluation and selection. Expert Syst. Appl. **37**(2), 1235–1240 (2010)
10. Parkouhi, S.V., Ghadikolaei, A.S.: A resilience approach for supplier selection: using fuzzy analytic network process and grey VIKOR techniques. J. Clean. Prod. **161**, 431–451 (2017)
11. Neto, P., Simão, M., Mendes, N., Safeea, M.: Gesture-based human-robot interaction for human assistance in manufacturing. Int. J. Adv. Manuf. Technol. **101**(1), 119–135 (2018). https://doi.org/10.1007/s00170-018-2788-x
12. Cwikla, G., Sekala, A., Wozniak, M.: The expert system supporting design of the manufacturing information acquisition system (MIAS) for production management. In: Advanced Materials Research, vol. 1036, pp. 852–857. Trans Tech Publications Ltd. (2014)
13. Zhang, J., Yang, Y., Zhuo, L., Tian, Q., Liang, X.: Personalized recommendation of social images by constructing a user interest tree with deep features and tag trees. IEEE Trans. Multimedia **21**(11), 2762–2775 (2019)
14. Lecouteux, B., Vacher, M., Portet, F.: Distant speech recognition in a smart home: comparison of several multisource ASRs in realistic conditions. In: Interspeech 2011 Florence, pp. 2273–2276 (2011)
15. Kulkarni, C.S., Bhavsar, A.U., Pingale, S.R., Kumbhar, S.S.: BANK CHAT BOT–an intelligent assistant system using NLP and machine learning. Int. Res. J. Eng. Technol. **4**(5), 2374–2377 (2017)
16. Shen, J., Wan, J., Lim, S.J., Yu, L.: Random-forest-based failure prediction for hard disk drives. Int. J. Distrib. Sens. Netw. **14**(11), 1550147718806480 (2018)
17. Kalsoom, A., Maqsood, M., Ghazanfar, M.A., Aadil, F., Rho, S.: A dimensionality reduction-based efficient software fault prediction using Fisher linear discriminant analysis (FLDA). J. Supercomput. **74**(9), 4568–4602 (2018). https://doi.org/10.1007/s11227-018-2326-5
18. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995). https://doi.org/10.1007/BF00994018
19. Breiman, L., Friedman, J., Olshen, R., Stone, C.: Cart. Classification and Regression Trees (1984)
20. Quinlan, J.R.: Induction of decision trees. Mach. Learn. **1**(1), 81–106 (1986). https://doi.org/10.1007/BF00116251
21. Salzberg, S.L.: C4. 5: programs for machine learning by J. Ross Quinlan. Morgan Kaufmann publishers, Inc., 1993 (1994)
22. Breiman, L.: Random forests. Mach. Learn. **45**(1), 5–32 (2001). https://doi.org/10.1023/A:1010933404324
23. Belhi, A., Bouras, A., Foufou, S.: Leveraging known data for missing label prediction in cultural heritage context. Appl. Sci. **8**(10), 1768 (2018). https://doi.org/10.3390/app8101768