ISSN: 2454-9940

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

E-Mail :
editor.ijasem@gmail.com
editor@ijasem.org

www.ijasem.org

INTERNATIONAL JOURNAL OF APPLIED SCIENCE ENGINEERING AND MANAGEMENT

ISSN2454-9940www.ijsem.org

Vol 9, Issuse.4 Oct 2021

The application of a Machine Learning Algorithm to a Model for Predicting Consumer Behavior

DR.D.B.K.KAMESH

Abstract –

The predictive power of machine learning algorithms has increased their relevance. Performance forecasting is challenging because of the unpredictability of individual client situations. There are several algorithms developed for the same task. In this study, we have examined the AODE, Naive Bayes, and AODEsr Bays algorithms. We used the WEKA tool to apply these methods and create a new model with higher accuracy. We've been working hard to clean up the data and make it more reliable throughout development, and now we need to filter out the irrelevant stuff. The newly filtered information will be given a weight of Wj as a result of this procedure. E (j, k), where j J or it's an assumption, defines the error. It all depends on your goals for k. N = E + Wj is another function that may be used to characterize noise.

INTRODUCTION

Machine learning is a part of artificial intelligence, in which we train a machine to predict the desired value. During this training, we define certain rules or patterns and our machine find out the defined pattern. So, in Machine Learning, input information is generated on the basis of knowledge stored in database. Since we are making our system to predict or extract relevant information from input data set, so, we need to develop an algorithm and pattern to retrieve the required information. After these two steps i.e. developments of algorithm and pattern have been completed, the machine can accomplish the

Following tasks:

Obtain, extract and summarize relevant information Make predictions based on analytical data Calculate the probability of certain effects to adapt to specific development independently

PROFESSOR, Mtech,Ph.D
Department of CSE
Gandhi Institute for Technology,Bhubaneswar.

TYPES OF MACHINE LEARNING

Machine learning algorithms are basically used to recognize patterns and subsequently generate a solution. Machine learning.algorithms.are classified as:

SUPERVISED LEARNING

In this type of learning, information is available in advance. In order to ensure adequate allocation of data to groups of algorithms, that should be explained. In other words, the system learns on the basis of input and output power. In supervised learning, the program manager, who acts as a type of teacher, gives the correct amount of feedback. The purpose is to train the method in the perspective of sequential input and output calculations and establish communication. The Naive Bayes is a model of probabilistic distinctions, based on the concept of autonomy. Though, in numerous real-world mining applications, this statement is often dishonored. In response to this statement, scholars have done a great deal of testing the correctness of NB by abating the quality of their stability. Webb et al. [1] have proposed an idea named Averaged One- Dependence Estimators (AODE) that decreases the independent predictive value by sampling all prototypes from a constrained class of dependent classifiers. Inspired by this research, we rely on that passing on diverse value to these different classifiers can lead to greater enhancement. We have experimentally verified our algorithm with Weak tool [2], using Super Market data sets and briefly defined a comparative study between Naive ayes, AODE and Aiders. The investigational outcomes indicate that proposed algorithm meaningfully leave behind all the other algorithms used to compare.

UNSUPERVISED LEARNING

In this learning scheme, values are not available previously. Basically it is used for clustering purposes. The machine tries to organize and filter the information entered according to specific features. For example, a machine can learn that coins of different colors can be arranged according to a different "color" to arrange them.

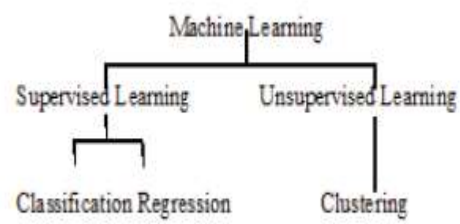

Fig. 1

NAIVEBAYS ALGORITHMS

Classification is first and foremost important thing in the area of data mining and machine learning. Learning about Bayesian classification is the process of making a different classifier from a training set by class labels.

Take Xi, i.=.1,.2,. ..., n,.are the values of the values xi, i.=.1,.2,...., n correspondingly. These are the attributes which will be used jointly to forecast the value of E of the study.

Therefore, the Bayesian classifier [9] can be demarcated as:

Arg.max.P(c) P(x1, x2, xn | c) where c.__C Consider that entire features are autonomous given the class, then the resultant classifier is named Naive ayes: arg.max P(c) _ _____ _ _ __ __where c_C

Correctly Classified Instances	2948	63.71%
Incorrectly Classified Instances	1679	36.29%
Kappa statistic	0	
Mean absolute error	0.4624	
Root mean squared error	0.4808	
Relative absolute error	100%	
Root relative squared error	100%	
Total Number of Instances	4627	
Time Taken to Build	0.02 seconds	

Fig. 2

Detailed Accuracy By Class			
			Weighted Avg.
TP Rate	1	0	0.637
FP Rate	1	0	0.637
Precision	0.637	0	0.406
Recall	1	0	0.637
F-Measure	0.778	0	0.496
ROC Area	0.499	0.499	0.499
Class	low	high	

Fig. 3

Confusion Matrix

a	b	← classified as
2948	0 |	a = low
1679	0 |	b = high

AODE

The most recent development project for NaiveBays is named Averaged One-Dependence Estimators, or simply AODE [1]. In AODE, a set of fixed-income students learn and a prediction is generated by guessing the guesses of all those students who once relied on one. For simplicity, a single dependency class is created first for each character, where the attribute is set to be the parent of all other attributes. Subsequently, AODE reaches directly to the aggregated scale containing most of the unique trees obtained by the Bayes construct. AODE divides the test model using Equation [9]:

$$\text{Arg} \quad \max\left(\sum_{i=1 \wedge T(x_i)\geq m}^{n} P(x_i, c) \prod_{j=1, j<>i}^{n} P(xj \mid x_i, c)\right/$$

!"#_$%&!') Where T (xi) is a calculation of the number of training sessions that have the value of the xi and is used to impose the limit on which they place on the support required to accept possible conditional limitations. The nonparent is the number of root symbols, fulfilling the condition that the training conditions contain more than m examples of the values of the parent attribute Ai. In the present study they use m = 30. In addition, AODE measures the probability basis P (xi, c) and P (xj | xi, c) as follows:

$$P(x_i, c) = T(a_i, c) + 1 / N + v_i * k$$
$$P(x_j \mid x_i, c) = T(x_j, x_i, c) + 1 / T(x_i, c) + v_j$$

The median reliability estimation algorithm works the same way as the Naive Bayesian class, but allows for two dimensional dependencies within the input test while continuing ignoring the complex dependency relationships Involving three or more values. AODE performs well with a large number of data objects. However, because all input price pairs are considered by the integrative method, it is not possible to use the AODE algorithm with high dimensional values. When there are multiple input values, it may be reasonable to use only dependency estimates in those cases where the dependency is proven or at least suspected.

Correctly Classified Instances	2957	63.91%
Incorrectly Classified Instances	1670	36.09%
Kappa statistic	0.0943	
Mean absolute error	0.492	
Root mean squared error	0.4923	
Relative absolute error	106.39%	
Root relative squared error	102.40%	
Total Number of Instances	4627	
Time Taken to Build	1.06 seconds	

Fig. 4

Detailed Accuracy By Class

			Weighted Avg.
TP Rate	0.9	0.182	0.639
FP Rate	0.818	0.1	0.558
Precision	0.659	0.507	0.604
Recall	0.9	0.182	0.639
F-Measure	0.761	0.268	0.582
ROC Area	0.687	0.68	0.687
Class	low	high	

Fig. 5

Confusion Matrix

```
   a      b            <- classified as
 2652   296 |           a = low
 1374   305 |           b = high
```

Subsumption Resolution (AODEsr) one-dimensional dependency ratios a certain type of dependency between Symbols results in a singular value of the other. For example, consider Gender and Pregnancy as two signs, and Baby = yes means Gender = woman. Therefore, gender = female is the width of pregnancy = yes. As such, Pregnancy = no standard sex = male. Where one value xi is a combination of the other, xj, P (y | xi, xj) = P (y | xj). As a result dumping the most common value from any calculation should not hurt any post merger equations, while assuming the independence between them may be. Motivated by this observation, Sub gumption Resolution (SR) [2] identifies two values so that one can appear to complete one and remove the norm.

Correctly Classified Instances	2959	63.95%
Incorrectly Classified Instances	1668	36.05%
Kappa statistic	0.0962	
Mean absolute error	0.4919	
Root mean squared error	0.4923	
Relative absolute error	106.38%	
Root relative squared error	102.39%	
Total Number of Instances	4627	
Time Taken to Build	2.23 Seconds	

Fig. 6

Detailed Accuracy By Class

			Weighted Avg.
TP Rate	0.899	0.184	0.64
FP Rate	0.816	0.101	0.557
Precision	0.659	0.509	0.605
Recall	0.899	0.184	0.64
F-Measure	0.761	0.27	0.583
ROC Area	0.687	0.687	0.687
Class	low	high	

Fig. 7

Confusion Matrix

```
   a      b      <- classified as
 2650   298 |  a = low
 1370   309 |    b = high
```

PROPOSED MODEL

Many inaccurate features may be existing in the data to be extracted. Therefore we need to identify and remove such type of inaccurate data. There are numerous mining procedures those are not good for large numbers of attributes. So, the feature selection techniques need to be used before the introduction of any type of mining algorithm. The basic purposes of feature selection are to simplify overload and optimize model performance and deliver quicker and more accurate representations. The biggest task with overloading and machine learning is that we don't know how well our model will work on new data until and unless we test it on the data set.. To do this, we can split our initial dataset into training and test subsets separately. We'll train and tune our model with training set and then will Apply to test set. Once the data is filtered, it will be stored in other file and will be allotted a weight wj. This data will be used to train our model. Data may come continuously in large volume. So, every data will go under this process and will be allotted with a constant value wj. An error can be demarcated as a function E(j, k) where j._.J or it is hypothesis and k is the goal function. Similarly noise can be defined by another function N= E + Wj.

ALGORITHM FOR PROPOSED MODEL

1: begin

2: Insert the data values with their attribute

3: Check the noise and find out the error rate

4: Filter the data on the basis of error rate and categories them.

5: Assign a weight Wi to filtered and corrected data

6: Use this weighted data to test a model

7: Select the model which is classifying correctly

8: end

	NaiveBays	AODE	AODEsr	Proposed Model
TP Rate	1	0.9	0.899	1
FP Rate	1	0.818	0.816	1
Precision	0.637	0.659	0.659	0.699
Recall	1	0.9	0.899	0.789
F-Measure	0.778	0.761	0.761	0.751
ROC Area	0.499	0.687	0.687	0.711
Correctly Classified Instances	0.63713	0.639075	0.639507	0.6425
Incorrectly Classified Instances	0.36287	0.360925	0.360493	0.3575

***Fig. 8** Comparison of NB, AODE, AODEsr and Proposed Model*

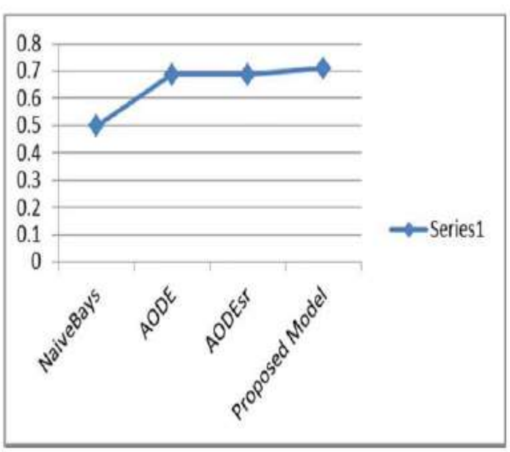

***Fig. 9** Graphical Representation of performance with Proposed Model*

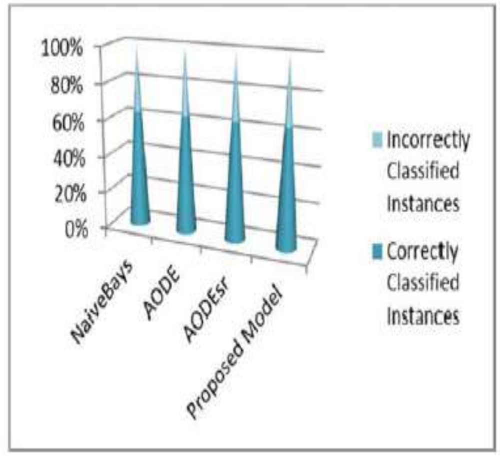

Fig. 10 %age of classification of NB, AODE, AODEsr and Proposed Model

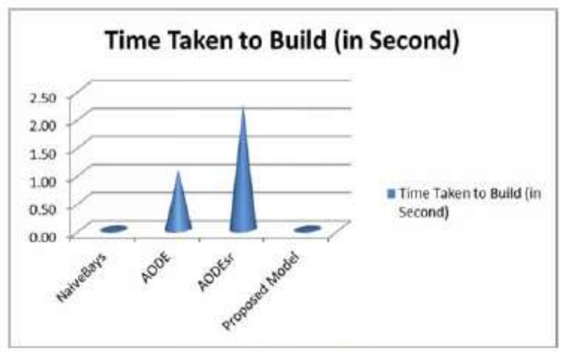

Fig. 11 Time taken to build the model

CONCLUSION

The experimental evidence suggests that the proposed method outperforms Naive Bays, AODE, and Aiders in terms of accuracy. Comparatively, it has a shorter learning curve than AODE and Aiders. The approach proposed in this study is a complicated one-dimensional model, and as such is not comparable to high-dimensional data...

REFERENCES

[1] Webb, G.I., and Bought on, J., Wang, and Z.: Not so naive bays: Aggregating one-dependence

estimators. Machine Learning 58, 5– 24 (2005) zbMATHCrossRefGoogle Scholar
[2] Chen S., Martinez A.M., Webb G.I. (2014) Highly Scalable Attribute Selection for Averaged One-Dependence Estimators. In: Tseng V.S., Ho T.B., and Zhou ZH. Chen A.L.P., Kao HY. (Ends) Advances in Knowledge Discovery and Data Mining. PAKDD 2014. Lecture Notes in Computer Science, vol 8444. Springer, Cham.
[3] Witten, I.H., Frank, E.: Data mining-Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, San Francisco (2000),http://prdownloads.sourceforge.net/weka/datasets-UCI.jar Google Scholar
[4] Pearl, J.: Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann, San Francisco (1988) Google Scholar
[5] Merz, C., Murphy, P., Aha, D.: UCI repository of machine learning databases. In Dept of ICS, University of California, Irvine (1997), http://www.ics.uci.edu/mlearn/MLRepository.html
[6] Langley, P., Sage, S.: Induction of selective Bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp. 339–406 (1994) Google Scholar
[7] Kohavi, R.: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD 1996), pp. 202–207. AAAI Press, Menlo Park (1996) Google Scholar
[8] Friedman, Geiger, Goldszmidt: Bayesian Network Classifiers. Machine Learning 29, 131–163 (1997) zbMATHCrossRefGoogle Scholar
[9] Jiang L., Zhang H. (2006) Weightily Averaged One-Dependence Estimators. In: Yang Q., Webb G. (eds) PRICAI 2006: Trends in Artificial Intelligence. PRICAI 2006. Lecture Notes in Computer Science, vol 4099. Springer, Berlin, Heidelberg.
[10] Chickering, D.M.: Learning Bayesian networks are NP-Complete. In: Fisher, D., Lenz, H. (eds.) Learning from Data: Artificial Intelligence and Statistics V, pp. 121–130. Springer, Heidelberg (1996) Google Scholar
[11] Fey Zheng, Geoffrey I. Webb: Efficient Lazy Elimination for Averaged-One Dependence Estimators. In: Proceedings of the Twenty-third International Conference on Machine Learning (ICML 2006), 1113-1120, 2006.