International Journal of Scientific Research and Engineering Development-– Volume 5 Issue 2, Mar-Apr 2022

Available at www.ijsred.com

RESEARCH ARTICLE OPEN ACCESS

Identification of Prospective User's with the use of Machine Learning Techniques

Dr. Vinod Sharma

Department of Computer Science Engineering and application, S.C.E. ,M.P. India,vs100880rs@gmail.com

Abstract :- Over the years, data creation and management methods have gone through some remarkable changes. Firstly, it was limited to organizations and enterprises, but with the advent of the Internet, those entities were enabled to share some of this data with others. The Huge flow of data required analytics for its better ment . Then, with the booming spread of social media, internet users were able to contribute and share data in different formats(e.g., videos on YouTube, tweets on Twitter, images on Instagram, etc.) Similarly, mobile devices exaggerated the amount of data being produced and shared. Add to this, the floods of data generated every day by machines, actuators and sensors (as described earlier in the "Internet of Things" section) and of course, scientific experiments and simulations that yield petabytes of data every day.In this research an approach implemented on big data to find out prospective customer.by probability measurement and also find out loyalty of customer through above algorithm implementation.

Keywords :- Axis ,Class, Events, Learning, Probability, Regression, Vector.

Related work :- Big data in information retrieval refers to electronic data sets so large and complex that they are difficult (or impossible) to manage with traditional software and/or hardware; nor can they be easily managed with traditional or common data management tools and methods.So, the different machine learning techniques are come in to picture. Firstly it is necessary to know how Big Data are collect, Steps to Collect Big Data.In step one gather data,. There are many ways to gather data according to different purposes. Here we apply our learning approaches.In next step which the storing data, after gathering the big data, you can put the data into databases or storage services for further processing.after this step requirement of Clean up data. In next requirement of reorganize data and in next verify data.

Big data can help predict equipment failure.With this data, manufacturers can maximize parts and equipment uptime and deploy maintenance more cost effectively. This data can be used to predict more than just equipment failure. For many manufacturing processes, it's also important to predict the remaining optimal life of systems and components to ensure that they perform within specifications. Falling out of tolerance—even if nothing is broken—can be as bad as failure. For example: in drug manufacturing a faulty, but still functional, component could introduce too much or too little of the active ingredient.

Operational efficiency is one of the areas in which big data can have the most impact on profitability. With big data, you can analyze and assess production processes, proactively respond to customer

feedback, and anticipate future demands. Potential issues can be discovered by analyzing both structured data (equipment year, make, and model) and multi-structured data (log entries, sensor data, error messages, engine temperature, and other factors).

Optimizing production lines can decrease costs and increase revenue. Big data can help manufacturers understand the flow of items through their production lines and see which areas can benefit. Data analysis will reveal which steps lead to increased production time and which areas are causing delays.

III. Proposed Methodology

In around the world most of user are trying to get their future purchasing item through internet. It means internet storage is a space where prospective customer are listed by their search. Big data can be used to improve the in-store experience. Many retailers are starting to analyze data from mobile apps, in-store purchases, and geo locations to optimize merchandizing encourage customers to complete purchases. With the help of machine learning techniques there is a possibelties to segregate this data according to user search and find out prospective customer for a particular product. In second implementation of another techniques the data are categorized and set probability ratio to analysis of loyalty of customer.

Retailers need to know the true profitability of their customers, how markets can be segmented, and the potential of any future opportunities. End-to-end profit and margin analysis can help with identifying pricing improvement opportunities and areas where profits may be leaking.

Machine learning has been increasing tremendously in recent years due to the high demand and advancements in technology. The potential of machine learning to create value out of data has made it appealing for businesses in many different industries.

Most machine learning products are designed and implemented with off-the-shelf machine learning algorithms with some tuning and minor changes.There is a wide variety of machine learning algorithms that can be grouped in three main categories:

1. Supervised learning algorithms model the relationship between features (independent variables) and a label (target) given a set of observations. Then the model is used to predict the label of new observations using the features. Depending on the characteristics of target variable, it can be a **classification** (discrete target variable) or a **regression** (continuous target variable) task.

2. Un supervised learning algorithms try to find the structure in unlabeled data.

3. Reinforcement learning works based on an action-reward principle. An **agent** learns to reach a goal by iteratively calculating the **reward** of its actions.

IV Praposed Techniques

(A). Linear Regression

Linear regression is a **supervised** learning algorithm and tries to model the relationship between a **continuous** target variable and one or more independent variables by fitting a linear equation to the data.For a linear regression to be a good choice, there needs to be a linear relation between independent variable(s) and target variable. There are many tools to explore the relationship among variables such as scatter plots and correlation matrix. For example, the scatter plot below shows a positive correlation between an independent variable (x-axis) and dependent variable (y-axis). As one increases, the other one also increases.

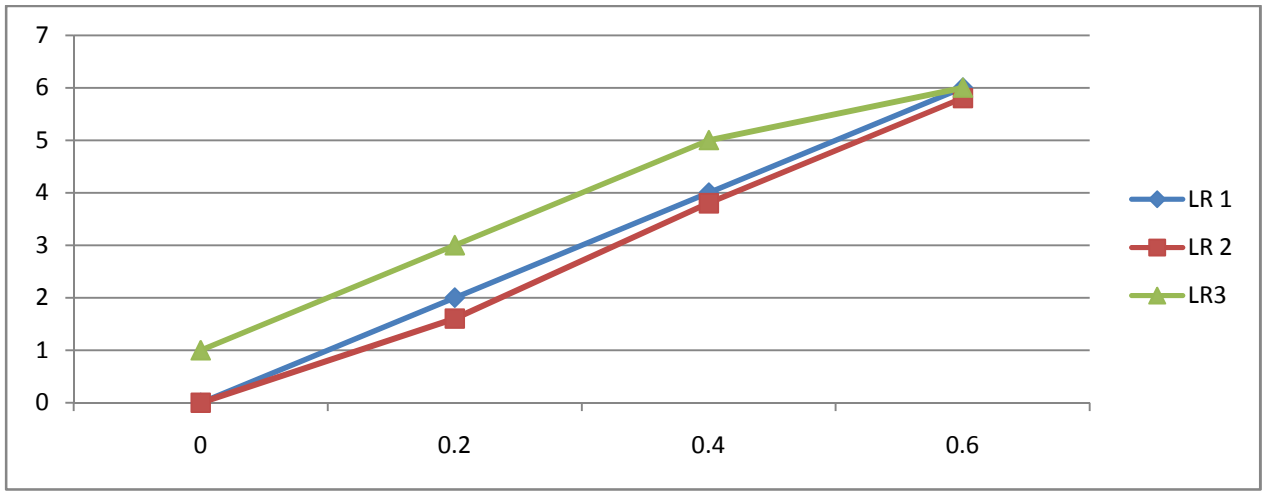

Fig-1. Lenear Regression correlation between axis

A linear regression model tries to fit a regression line to the data points that best represents the relations or correlations. The most common technique to use is **ordinary-least squares** (OLE). With this method, best regression line is found by minimizing the sum of squares of the distance between data points and the regression line. For the data points above, the regression line obtained using OLE seems like:

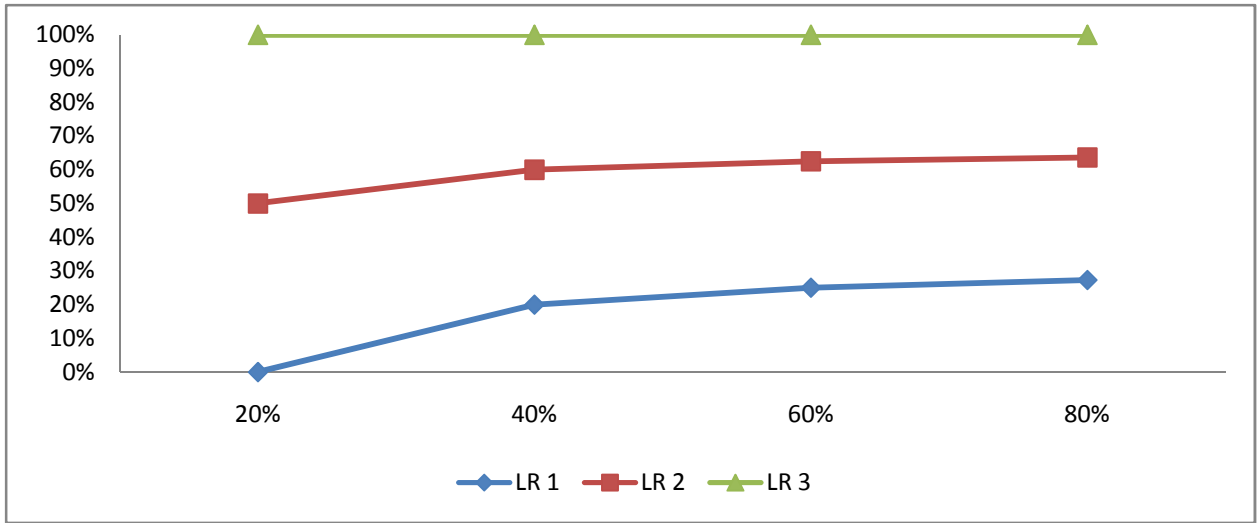

Figure-2 Obtained regression line using OLE.

Linear Regression is an ML algorithm used for supervised learning. Linear regression performs the task to predict a dependent variable(target) based on the given independent variable(s). So, this regression technique finds out a linear relationship between a dependent variable and the other given independent variables.

In the figure above, on X-axis is the independent variable and on Y-axis is the output. The regression line is the best fit line for a model. And our main objective in this algorithm is to find this best fit line.

Pros:

- Linear Regression is simple to implement.
- Less complexity compared to other algorithms.
- Linear Regression may lead to over-fitting but it can be avoided using some dimensionality reduction techniques, regularization techniques, and cross-validation.

Cons:

- Outliers affect this algorithm badly.
- It over-simplifies real-world problems by assuming a linear relationship among the variables, hence not recommended for practical use-cases.
-
- **Support Vector Regression**

You must have heard about SVM i.e., Support Vector Machine. SVR also uses the same idea of SVM but here it tries to predict the real values. This algorithm uses hyperplanes to segregate the data. In case this separation is not possible then it uses kernel trick where the dimension is increased and then the data points become separable by a hyperplane.

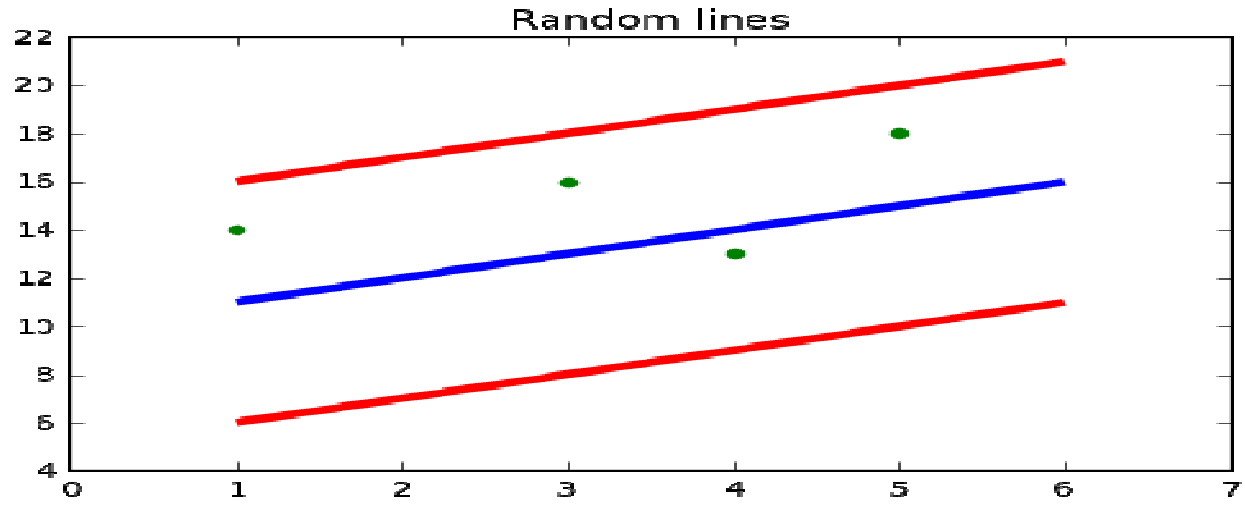

Figure-3 Predict the real values using SVM

In the figure above, **the Blue line is the Hyper Plane; Red Line is the Boundary Line**

All the data points are within the boundary line(Red Line). The main objective of SVR is to basically consider the points that are within the boundary line.

Some Pros are following

1. Robust to outliers.

2. Excellent generalization capability

3. High prediction accuracy.

Some Cons are following:

1 . Not suitable for large datasets.

2. They do not perform very well when the data set has more noise.

Then the need of classification required on space area data which are essentially search by different users. Data may be stored in different storage by data analytics. These data are processed for next level where implementation of naïve bayes supervised learning are perform. Naive bayes Algorithm are find out the probability accuracy of selection items from among item set of data.

(B) Naive Bayes

Naive Bayes is a **supervised** learning algorithm used for classification tasks. Hence, it is also called Naive Bayes Classifier. p(A|B): Probability of event A given event B has already occurred

p(B|A): Probability of event B given event A has already occuured

p(A): Probability of event A

p(B): Probability of event B

Naive bayes classifier calculates the probability of a class given a set of feature values (i.e. p(yi | x1, x2 , … , xn)). Input this into Bayes' theorem:

$$p(yi \mid x1, x2, \ldots, xn) = \frac{p(x1, x2, \ldots, xn \mid yi).p(yi)}{p(x1, x2, \ldots, xn)}$$

p(x1, x2 , … , xn | yi) means the probability of a specific combination of features (an observation / row in a dataset) given a class label. We need extremely large datasets to have an estimate on the probability distribution for all different combinations of feature values. To overcome this issue, **naive bayes algorithm assumes that all features are independent of each other.** Furthermore, denominator (p(x1,x2, … , xn)) can be removed to simplify the equation because it only normalizes the value of conditional probability of a class given an observation (p(yi | x1,x2, … , xn)).The probability of a class (p(yi)) is very simple to calculate:

$$p(yi) = \frac{\text{number of observations with class yi}}{\text{number of all observations}}$$

Under the assumption of features being independent, **p(x1, x2 , … , xn | yi)** can be written as**:**

$$p(x1, x2, \ldots, xn \mid yi) = p(x1 \mid yi) \,.\, p(x2 \mid yi) \,.\, \ldots \,.\, p(xn \mid yi)$$

The conditional probability for a single feature given the class label (i.e. p(x1 | yi)) can be more easily estimated from the data. The algorithm needs to store probability distributions of features for each class independently. For example, if there are 5 classes and 10 features, 50 different probability distributions need to be stored.Adding all these up, it became an easy task for naive bayes algorithm to calculate the probability to observe a class given values of features (**p(yi | x1, x2 , … , xn)**)The assumption that all features are independent makes naive bayes algorithm **very fast** compared to complicated algorithms. In some cases, speed is preferred over higher accuracy. On the other hand, the same assumption makes naive bayes algorithm less accurate than complicated algorithms. Speed comes at a cost.Naive bayes assumes that **features are independent of each other** and **there is no correlation between features**. However, this is not the case in real life. This naive assumption of features being uncorrelated is the reason why this algorithm is called “naive”.

$$p(A|B) = \frac{p(A).p(B|A)}{p(B)} \qquad (Bayes' \ Theorem)$$

p(A|B):

Probability of event A given event B has already occurred

p(B|A): Probability of event B given event A has already occuured

p(A): Probability of event A

p(B): Probability of event B

Naive bayes classifier calculates the probability of a class given a set of feature values (i.e. p(yi | x1, x2 , … , xn)). Input this into Bayes’ theorem:

$$p(yi \mid x1, x2, \ldots, xn) = \frac{p(x1, x2, \ldots, xn \mid yi).p(yi)}{p(x1, x2, \ldots, xn)}$$

p(x1, x2 , … , xn | yi) means the probability of a specific combination of features (an observation / row in a dataset) given a class label. We need extremely large datasets to have an estimate on the probability distribution for all different combinations of feature values. To overcome this issue, **naive bayes algorithm assumes that all features are independent of each other.** Furthermore, denominator (p(x1,x2, … , xn)) can be removed to simplify the equation because it only normalizes the value of conditional probability of a class given an observation (p(yi | x1,x2, … , xn)).

The probability of a class (p(yi)) is very simple to calculate:

$$p(yi) = \frac{\text{number of observations with class yi}}{\text{number of all observations}}$$

Under the assumption of features being independent, **p(x1, x2 , … , xn | yi)** can be written as**:**

$$p(x1, x2, \dots, xn \mid yi) = p(x1 \mid yi) \cdot p(x2 \mid yi) \cdot \dots \cdot p(xn \mid yi)$$

The conditional probability for a single feature given the class label (i.e. p(x1 | yi)) can be more easily estimated from the data. The algorithm needs to store probability distributions of features for each class independently. For example, if there are 5 classes and 10 features, 50 different probability distributions need to be stored.

Adding all these up, it became an easy task for naive bayes algorithm to calculate the probability to observe a class given values of features (**p(yi | x1, x2 , … , xn))**

The assumption that all features are independent makes naive bayes algorithm **very fast** compared to complicated algorithms. In some cases, speed is preferred over higher accuracy. O`n the other hand, the same assumption makes naive bayes algorithm less accurate than complicated algorithms. Speed comes at a cost.

Experiments and Results Analysis:-

First we will develop each piece of the algorithm in this section, then we will tie all of the elements together into a working implementation applied to a real dataset in the next section.

This Naive Bayes tutorial is broken down into 5 parts:

Step 1: Separate By Class.
Step 2: Summarize Dataset.
Step 3: Summarize Data By Class.
Step 4: Gaussian Probability Density Function.

Step 5: Class Probabilities.

These steps will provide the foundation that you need to implement Naive Bayes from scratch and apply it to your own predictive modeling problems.

```
def  separate _ by _class(dataset):
        separated =  dict ()
        for i in range ( len (dataset)):
                vector = dataset[i]
                class _value = vector[-1]
                if (class _value not in separated):
                        separated[class _value] = list()
                separated[class _value].append(vector)
        return  separated
```

Need to calculate the probability of data by the class they belong to, the so-called base rate.

This means that we will first need to separate our training data by class. A relatively straightforward operation.

We can create a dictionary object where each key is the class value and then add a list of all the records as the value in the dictionary.

Function named *separate_by_class()* that implements this approach. It assumes that the last column in each row is the class value.

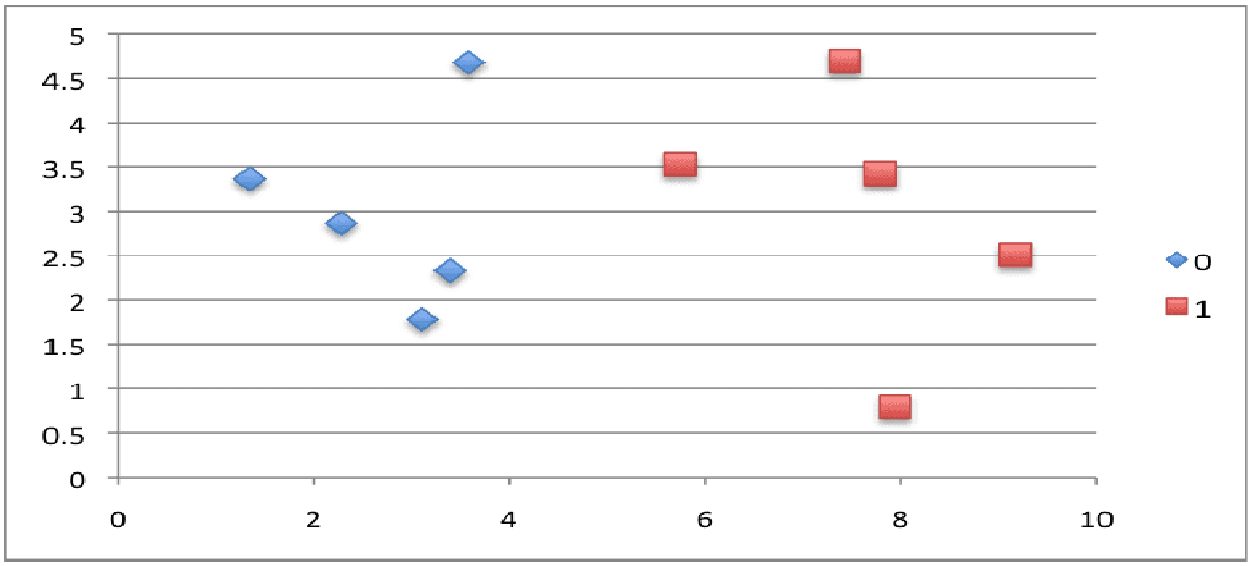

Figure-4 Diagram o separation by class

```
def  separate_by_class (dataset):
        separated = dict()
        for  i in range (len(dataset)):
                vector = dataset[i]
```



```
            class_ value = vector[-1]
            if (class _value not in separated):
                  separated [ class _value] = list()
            separated[class _value].append (vector)
      return separated

# Test separating data by class
Separated  = separate_ by _ class(dataset)
for  label in separated:
      print  (label)
      for row in separated[label]:
            print (row)
```

RESULT ANALYSIS :-with above experiments separated data are retrieve through different techniques of machine learning and stored data seprated according to different label and define different prospective customer's for specified products. Further repeatation of algorithm regratively find out consecutive behavior of search so, this can be fruitful for loyalty of user also. As the increasing scenario of data usage and distribution, this implementation are generate close analysis and accurate probability of prospective user.

CONCLUSIONS:- As time progresses , marketism will also increase rapidly and commercial competition in the market will also increase rapidly.As the time being demands of market are increasing regularly and there is compulsory to identification of prospective Costomers.With the help of the machine learning techniques,the prospective customer can be searched from the bottomless crowd(data),where only they has to make only effort to find his favorite item with the help of any medium of internet and this effort of his reach his search through this technology. As well as giving him a lot of options for his favorite item. Above process will continue to decorate the future in a new form with such new technologies and creations.

WEB REFERENCES :-

[1] https://www.oracle.com/a/ocom/docs/top-22-use-cases-for-big-data.pdf

[2] https://machinelearningmastery.com/naive-bayes-classifier-scratch-python

[3] https://www.oracle.com/big-data/guide/big-data-use-cases.html

[4] https://www.aiproblog.com/index.php/2019/10/17/naive-bayes-classifier-from-scratch-in-python

[5] https://towardsdatascience.com/naive-bayes-classifier-explained-50f9723571ed

[6] https://www.engati.com/glossary/machine-learning-algorithms

[7] https://towardsdatascience.com/a-beginners-guide-to-supervised-machine-learning-algorithms-6e7cd9f177d5

[8] https://towardsdatascience.com/11-most-common-machine-learning-algorithms-explained-in-a-nutshell-cc6e98df93be

[9] https://www.analyticsinsight.net/types-of-machine-learning-algorithms-one-should-know-about/

REFERENCES:-

[1] Arbeláez, P.A., Girshick, R.B., Hariharan, B., & Malik, J. (2014) ECCV , (cited 286 times, HIC: 23 , CV: 94)

[2] Chandrashekar , G., & Sahin, F. Int. J. on Computers & Electrical Engineering, (cited 279 times, HIC: 1 , CV: 58).

[3]. Chan YH. Biostatistics 103: Qualitative data – Tests of independence. Singapore Med J 2003;44:498-503.

[4] Freedman DA. Statistical Models: Theory and Practice. Cambridge, USA: Cambridge University Press; 2009. 3. Chan YH. Biostatistics 201: Linear regression analysis. Age (years). Singapore Med J 2004;45: 55-61.

[5]. Gaddis ML, Gaddis GM. Introduction to biostatistics: Part 6, correlation and regression. Ann Emerg Med 1990;19:1462-8.

[6] I. Rish, J. Hellerstein, and T. Jayram. An analysis of data characteristics that affect naive Bayes performance. Technical Report RC21993, IBM T.J. Watson Research Center, 2001.

[7] J. Hellerstein, Jayram Thathachar, and I. Rish. Recognizing end-user transactions in performance management. In Proceedings of AAAI-2000, pages 596–602, Austin, Texas, 2000.

[8] J. Hilden. Statistical diagnosis based on conditional independence does not require it. Comput. Biol. Med., 14(4):429–435, 1984.

[9]. Mendenhall W, Sincich T. Statistics for Engineering and the Sciences. 3rd ed. New York: Dellen Publishing Co.; 1992.

[10] N. Friedman, D. Geiger, and Goldszmidt M. Bayesian network classifiers. Machine Learning, 29:131–163, 1997.

[11]. Panchenko D. 18.443 Statistics for Applications, Section 14, Simple Linear Regression. Massachusetts Institute of Technology: MIT Open Course Ware; 2006.

[12] P. Langley, W. Iba, and K. Thompson. An analysis of Bayesian classifiers. In Proceedings of the Tenth National Conference on Artificial Intelligence, pages 399–406, San Jose, CA, 1992. AAAI Press.

[13] R.O. Duda and P.E. Hart. Pattern classification and scene analysis. New York: John Wiley and Sons, 1973.

[14] R. Kohavi. Wrappers for performance enhancement and oblivious decision graphs. Technical report, PhD thesis, Department of Computer Science, Stanford, CA, 1995.

[15] Tom M. Mitchell. Machine Learning. McGraw-Hill, 1997.

