Determining the best classifier for predicting the value of a boolean field on a blood donor database using genetic algorithms

Ngô Thị Thùy Lam, Nguyễn Ngọc Thanh Trúc, Nguyễn Cao Bằng
TP. Hồ Chí Minh, Việt Nam, lamntt21@uef.edu.vn
TP. Hồ Chí Minh, Việt Nam, trucnnt21@uef.edu.vn
TP. Hồ Chí Minh, Việt Nam, bangnc21@uef.edu.vn

Abstract:

We frequently have access to sizable databases made up of several types of information, such as words, numbers, and even boolean values, thanks to digitization.

These databases are particularly well suited for applications involving machine learning, categorization, and large data analysis. particular that we know the values of the other fields, we may use classifiers that have been trained on already collected data to predict the values of a particular field.

In this work, we focus particularly on hospital-generated Electronic Health Records (EHRs). Although these EHRs provide easy access to patient data for specific individuals, processing that data as a whole is still difficult. EHRs that are made up of coherent, well-tabulated structures, on the other hand, provide excellent candidates for applying machine language through the use of classifiers. We examine a data set from a blood transfusion service center in this study (data sourced from a blood transfusion service center in Taiwan's Hsin-Chu City). We employed Python's scikit-learn machine learning library. We employ Support Vector Classification (SVC) from Support Vector Machines (SVM), and we import Perceptron from the linear model. The K.neighborsclassifier and decision tree classifiers were also utilized. Additionally, we employ evolutionary algorithms to find an optimum pipeline utilizing the TPOT library. We rate each of the aforementioned classifiers using k fold cross-validation.

Results:

The test program depends on evaluating each classifier separately. It displays these numbers and counts the number of forecasts that are close to the true value. We may choose the best classifier for the given blood donor database using the counts. We will be able to choose the most precise prognosis for each patient using the most accurate models, or a combination of these models. In this case, we're looking to see if a patient donated blood in March 2017. A boolean value of 1 or 0 represents this prediction, with 1 indicating that the patient had donated blood and 0 indicating otherwise.

Keyword: Boolean, Blood donor, Classifiers

1. Introduction

The ability to store data on electronic media has drastically improved over the past 20 years. The amount of medical data saved on electronic media has grown tremendously as a result. We have access to a variety of medical data, including text, photos, audio, and video. One of the few data types that may be found and used in medical facilities is this one. The necessary data is often used and examined on an individual basis. For instance, an analysis of an MRI and a textual health record will be done to determine a patient's diagnosis or the progression of their ailment.

Only structured data can be easily utilized to train machine learning algorithms out of all the other sorts of data. This is because tabular data makes for very effective machine language classifier training. As long as we are aware of the values present in the other fields of the dataset, we may use classifiers to analyze a dataset and predict the values present in one field of the supplied set. These machine learning algorithms work by identifying patterns in a dataset and applying those patterns to forecast values for fields that are missing data.

The majority of information in medical databases, including patient health charts and electronic patient reports, are unstructured free texts, which presents a difficulty. These speak natural languages, which are comprehended and processed more readily by people than by machines. These could contain prescription notes, notes about a patient's diagnosis, or even notes about a tissue sample. The most accurate pattern reconstruction and predictions in medicine are accomplished utilizing structured databases and classifier algorithms, even if NLP techniques may make free text machine comprehensible.

The program's technological foundation is based on the idea that different training methods, when trained on the same dataset, have varying success rates. We will then be able to decide which classifier pairs well with a certain method.

Although a doctor's diagnosis is still extremely useful in medicine, data science makes illness diagnosis and prognosis far more accurate and effective. Early identification, analysis, and use of algorithms as illness predictors aid physicians in disease analysis while also lowering patient mortality.

In this task, we will categorize whether a donor has given blood at a specific period (in this example, March 2007), based on the following criteria:

- How recently (in terms of months) a prior blood donation was made.
- How frequently people donate blood
- The cubic centimeters (cm^3) of blood donated
- The number of months from the initial donation.

A classifier's hyperparameters will also be tuned using a genetic algorithm, and its accuracy score will be compared to that of the default classifiers.

2. Methods

Binary classification and multiclass classification can be seen as two distinct challenges in classification. Multiclass classification entails placing an item in one of multiple classes, whereas binary classification, a simpler operation, just includes two classes.

While we classify an item into one of two categories when using binary classification, we classify an object into one of several categories when using multiclass classification. Our classifier gives us a boolean output of 1 or 0, where 1 indicates that the patient has contributed blood and 0 indicates otherwise, because our particular challenge requires us to determine if a certain patient had donated blood on a prior date.

Therefore, we have a binary classification problem to solve. The same dataset should be used to train each classifier in order to determine which one is the most effective. Then, for a dataset of brand-new patients, we should forecast the values 1 or 0, and see if the predictions correspond.

We can select the best effective classifier by calculating the percentage of matches in each database using the number of matched predictions.

A classification issue in machine learning and statistics is one of determining to which of a set of categories (sub-populations) a new observation belongs using a training set of data that contains observations (or instances) with known category membership.

Classification is a form of supervised learning, or learning where a training set of properly recognized observations is provided, according to the language of machine learning[1]. Clustering, the comparable unsupervised method, involves categorizing data based on some metric of innate similarity or distance.

2.1. Binary classification

A collection of measurable traits, sometimes referred to as explanatory variables or features, are frequently derived from the analysis of the individual data. These characteristics can be categorical (like "A," "B," "AB," or "O" for blood type), ordinal (like "large," "medium," or "small"), integer-valued, or real-valued (like a blood pressure reading). data are compared to prior data using a similarity or distance function in other classifiers.

A classifier is an algorithm that executes classification, particularly in a practical implementation. The mathematical operation carried out by a classification algorithm that assigns input data to a category is sometimes referred to as a "classifier" on occasion.

Term use varies a lot between fields. The characteristics of observations are known as explanatory variables (or independent variables, regressors, etc.) in statistics, where classification is frequently done with logistic regression or a similar procedure, and the categories to be predicted are known as outcomes, which are considered to be possible values of the dependent variable. In machine learning, the various categories that may be predicted are referred to as classes, the observations are frequently referred to as instances, and the explanatory factors are referred to as features (grouped into a feature vector). Different terms may be used in other disciplines. For instance, in community ecology, the term "classification" typically refers to cluster analysis, a kind of unsupervised learning, rather than the supervised learning discussed in this article.

In binary classification, we group an object into one of two classes. Our specific problem requires us to decide whether a specific patient had donated blood on a previous date, and as a result, our classifier provides us with a boolean output of either 1 or 0, wherein 1 denotes that the patient had donated blood and 0 denotes otherwise.

Thus, we have to solve a binary classification problem. In order to find the most efficient classifier, we should train each classifier on the same dataset. Then, we should predict the values 1 or 0 for a dataset of new patients and determine whether the predictions match. By finding out the number of matched predictions, we can calculate the percentage matches in each database and subsequently determine the most efficient classifier.

2.2. Classifier selection

Via the use of the classifier selection diagram given here: We have subsequently followed the steps:

- >50 samples: Yes
- Predicting a category : Yes
- Labeled Data : Yes
- <100k samples : Yes

Thus, we have chosen the following classifiers:

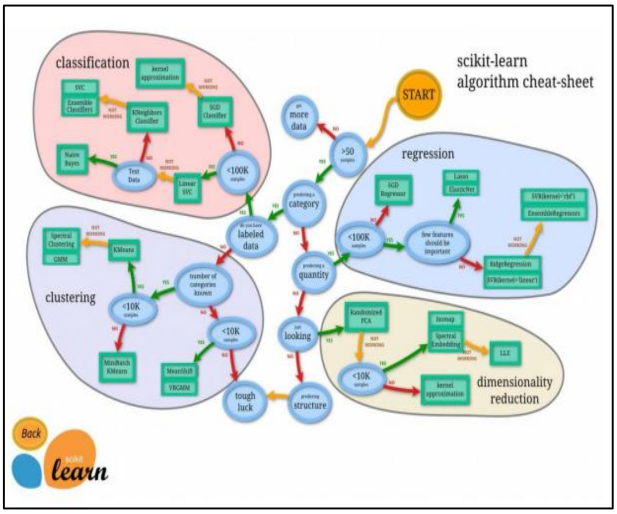


Figure 1: Approach problems about which estimators to try on data.

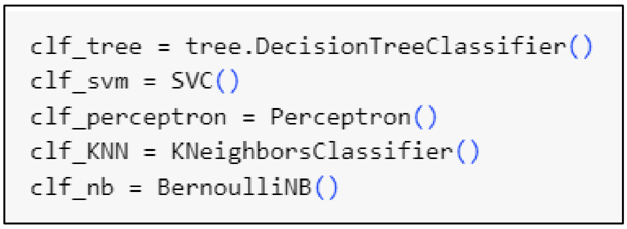
1. SVC

2. Perceptron

3. KneighborsClassifier

4. Decision Tree Classifier

5. Furthermore, we include a Nave Bayes classifier in our tests because of its high accuracy in binary classification.



```
clf_tree = tree.DecisionTreeClassifier()
clf_svm = SVC()
clf_perceptron = Perceptron()
clf_KNN = KNeighborsClassifier()
clf_nb = BernoulliNB()
```

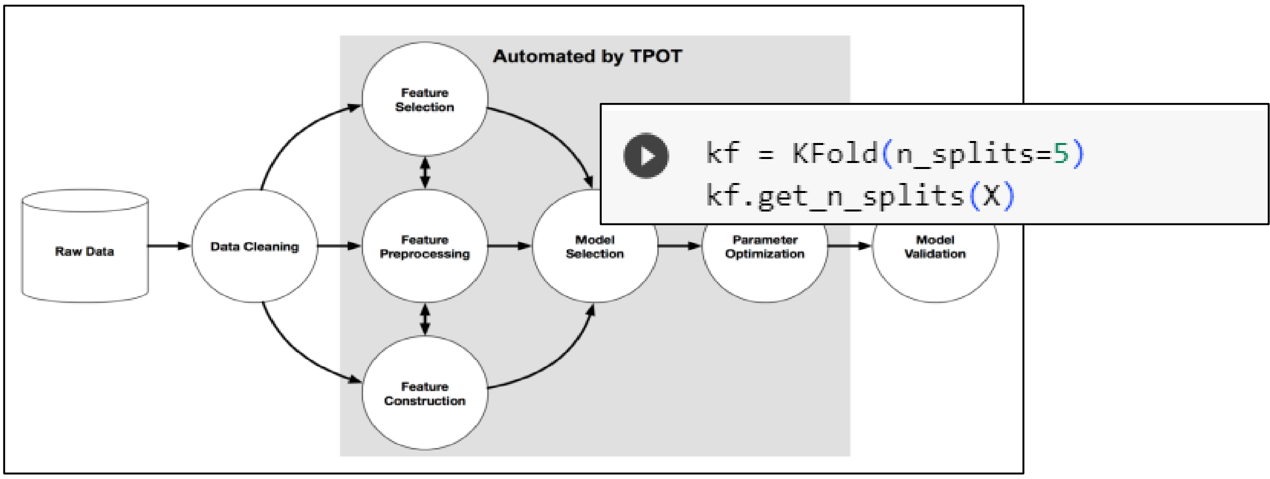
Figure 2: Use pseudocode to assign to algorithms

2.3. TPOT auto machine learning

We will also be training a classifier from the TPOT library to choose the best classifier with respect to accuracy, perform hyperparameter tuning on the said classifier, and discover the best pipeline.

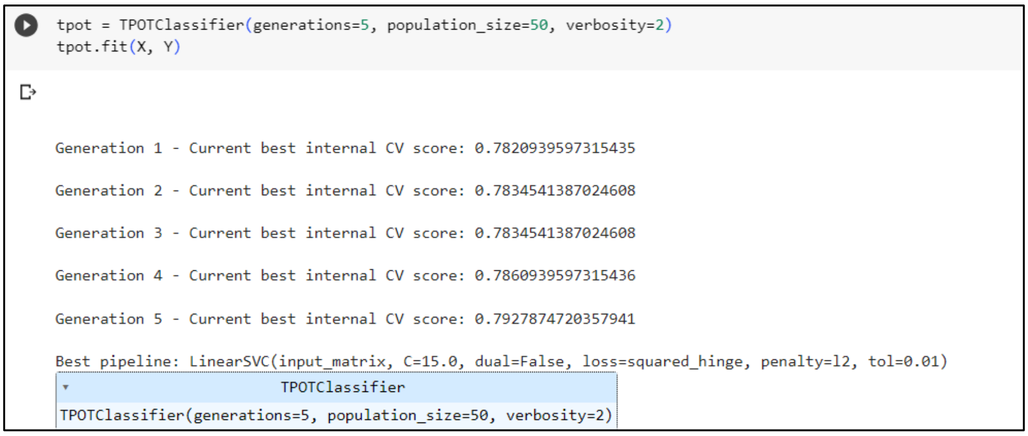
We will limit our discussion to the workings of the TPOT classifier, as the other classifiers are standard algorithms, and their implementation has been rigorously documented and discussed in the Scikit-lLearnPaper (see references). This is the automation flow of the TPOT library. It is built on top of Scikit-learn and performs hyperparameter tuning on Scikit-learn's regressors and classifiers.

A TPOT classifier will work with thousands of pipelines and then recommend the pipeline that works best for the given data. It applies genetic algorithms on antonitial population of pipelines to pick the most fit pipelines, reproduce a new generation of pipelpipelines,iterate this process over a number of generations. This process usually converges to a best pipeline, unless it is explicitly terminated by a programmer



```
kf = KFold(n_splits=5)
kf.get_n_splits(X)
```

Figure 3: An Example machine learning pipeline



```
tpot = TPOTClassifier(generations=5, population_size=50, verbosity=2)
tpot.fit(X, Y)
```

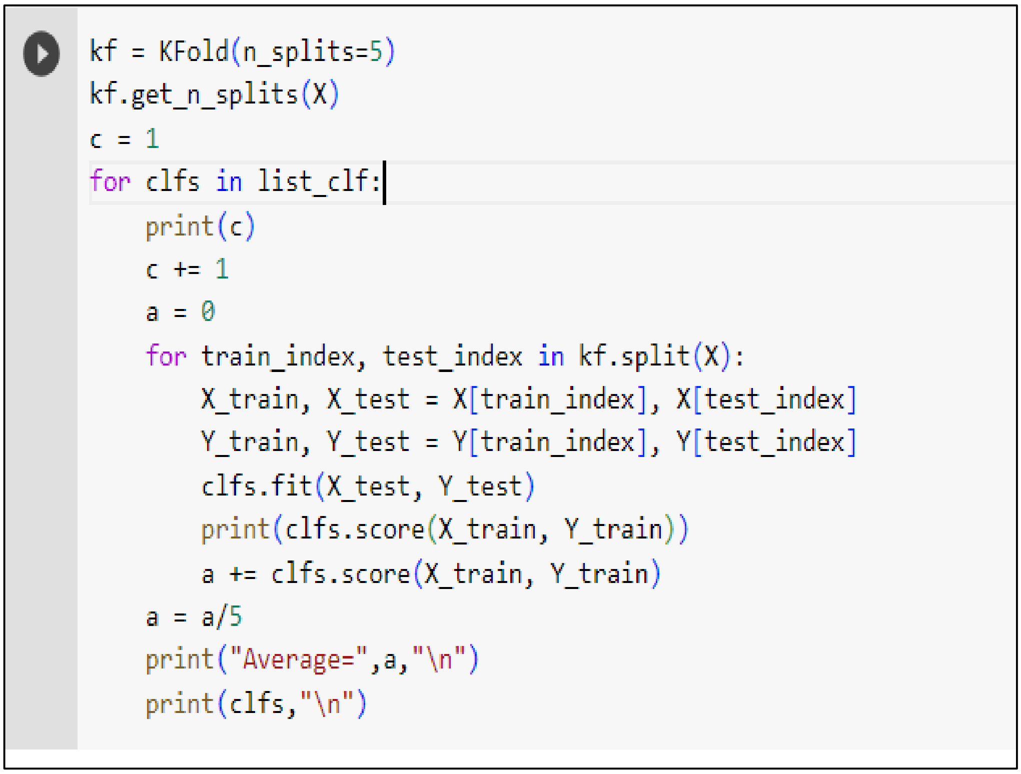
```
Generation 1 - Current best internal CV score: 0.7820939597315435

Generation 2 - Current best internal CV score: 0.7834541387024608

Generation 3 - Current best internal CV score: 0.7834541387024608

Generation 4 - Current best internal CV score: 0.7860939597315436

Generation 5 - Current best internal CV score: 0.7927874720357941

Best pipeline: LinearSVC(input_matrix, C=15.0, dual=False, loss=squared_hinge, penalty=l2, tol=0.01)
▾                    TPOTClassifier
TPOTClassifier(generations=5, population_size=50, verbosity=2)
```

Figure 4: TPOT Classifier model

Thus, the final list of the classifiers used is:

- SVC
- Perceptron
- KneighborsClassifier

- Decision Tree Classifier



```
kf = KFold(n_splits=5)
kf.get_n_splits(X)
c = 1
for clfs in list_clf:
    print(c)
    c += 1
    a = 0
    for train_index, test_index in kf.split(X):
        X_train, X_test = X[train_index], X[test_index]
        Y_train, Y_test = Y[train_index], Y[test_index]
        clfs.fit(X_test, Y_test)
        print(clfs.score(X_train, Y_train))
        a += clfs.score(X_train, Y_train)
    a = a/5
    print("Average=",a,"\n")
    print(clfs,"\n")
```

Figure 6: Determine accuracy score of each model.

- Nave Bayes Classifier
- TPOT Classifier

2.4. Cross validation

K-Fold is a method for splitting a dataset into k equal-sized folds. It is commonly used in machine learning to evaluate the performance of a model on a limited data sample. The procedure involves splitting the data into k groups and training the model on k-1 groups while using the remaining group as the test set. This process is repeated k times, with each group being used as the test set exactly once. The results are then averaged across all k iterations to produce a single performance estimate.

We chose n_splits = 5, that means,K-Fold splits the dataset into 5 folds, then trains on 4 folds and tests on the other one. This process would be iterate 5 times.

Figure 5: K-fold cross validation

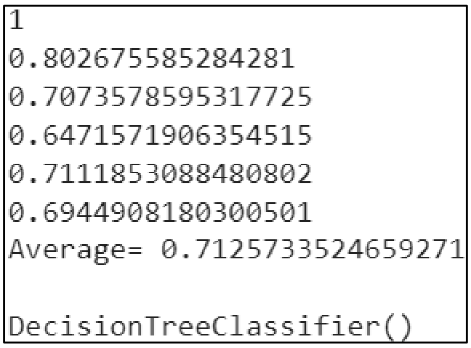
2.5. Evaluation

We evaluated each of the classifiers via k-fold cross-validation and determined the best classifier on the basis of the best average score.

We used a for loop to iterate over each classifier in the list of classifier models. Then, for each classifier, it iterates over each fold of the data, trains the classifier on the training set, and tests it on the test set. It then prints out the score of the classifier on training set 1. Finally, it calculates the average score across all folds and prints it out.

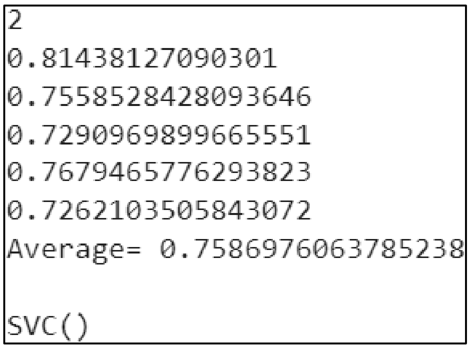
3. Results

The classifiers' accuracy ratings are as follows (notice that these output numbers were taken straight from the Python console).



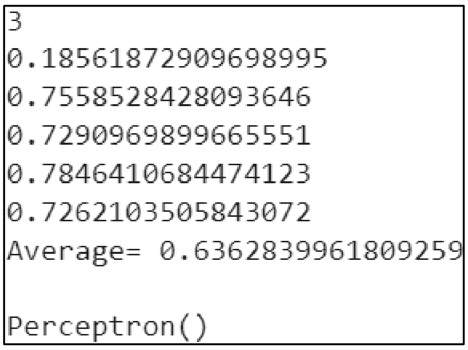
```
1
0.802675585284281
0.7073578595317725
0.6471571906354515
0.7111853088480802
0.6944908180300501
Average= 0.7125733524659271

DecisionTreeClassifier()
```

Figure 7: Decision Tree's result



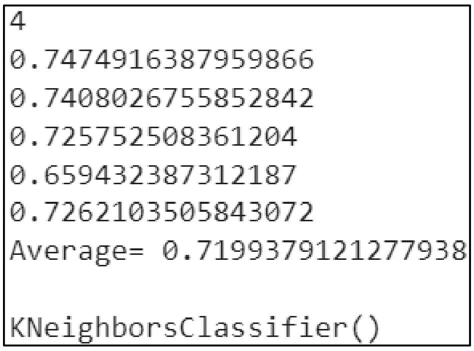
```
2
0.81438127090301
0.7558528428093646
0.7290969899665551
0.7679465776293823
0.7262103505843072
Average= 0.7586976063785238

SVC()
```

Figure 8: SVC's result



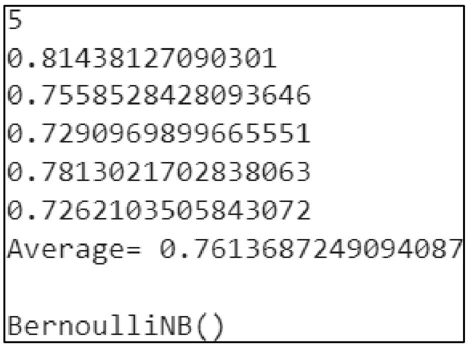
```
3
0.18561872909698995
0.7558528428093646
0.7290969899665551
0.7846410684474123
0.7262103505843072
Average= 0.6362839961809259

Perceptron()
```

Figure 9: Perceptron's result



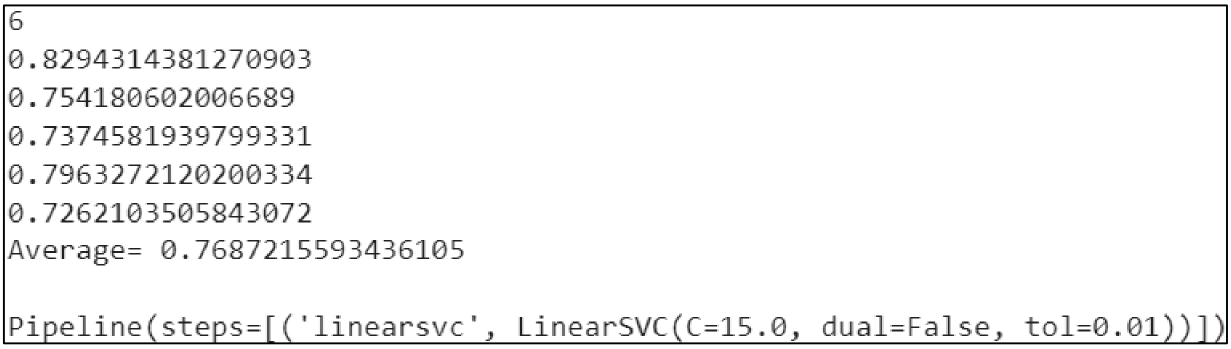
```
4
0.7474916387959866
0.7408026755852842
0.725752508361204
0.659432387312187
0.7262103505843072
Average= 0.7199379121277938

KNeighborsClassifier()
```

Figure 10: KNN's result



```
5
0.81438127090301
0.7558528428093646
0.7290969899665551
0.7813021702838063
0.7262103505843072
Average= 0.7613687249094087

BernoulliNB()
```

Figure 11: Naive Bayes's result



```
6
0.8294314381270903
0.754180602006689
0.7374581939799331
0.7963272120200334
0.7262103505843072
Average= 0.7687215593436105

Pipeline(steps=[('linearsvc', LinearSVC(C=15.0, dual=False, tol=0.01))])
```

Figure 12: LinearSVC's result

We find that the Nave Bayes Classifier comes in second with the best average score, followed by the TPOT optimized pipeline. In this instance, the LinearSVC classifier with adjusted hyperparameters serves as the TPOT-optimized pipeline.

(Note: Only five generations of the TPOT classifier fit optimization approach have been executed. It is possible to improve accuracy scores by increasing the number of generations.

We may draw a conclusion from this by noting that genetically enhanced pipelines perform best in this categorization job. On a k-fold cross-validation test, the Naive Bayes classifier has a comparable result, nonetheless.

4. Additional content

The two algorithms considered the best in this analysis, NB and LinearSVC, are not the latest techniques for the following reasons:

The above results are only based on one dataset, and the number of samples in that dataset is only about 200.

The above results are only based on one cross-validation method, KFold; maybe other cross-validation methods can bring better results for the models.

For the purpose of determining the best classification model for predicting the value of the Boolean field, the author should make more comparisons to increase the reliability of the final model. In a research paper with similar content: *In "the Computational Intelligibility of Boolean Classifiers by Gilles Audemard, Steve Bellart, Louenas Bounia, Frédéric Koriche, Jean-Marie Lagniez, and Pierre Marqui"*, the authors used some other models such as DNF formulae, decision lists, boosted trees, and binarized neural nets. In the future, we may apply the Naive Bayes technique in this article to my work. The research that we think of will be document classification, articles in the electronic library. The researches for Vietnamese language mainly focus on the problems of word separation (Word segmentation), type word recognition (Pos tagging) , the parser syntax (syntax analysis). Need for simplicity, unnecessary parameters are too large like other methods, flexibility for training training information changes, classification time in accordance with requirements, Naive Bayes appears very suitable with the setting request.

5. References:

[1]. Har-Peled, S., Roth, D., Zimak, D. (2003) "Constraint Classification for Multiclass Classification and Ranking." In: Becker, B., Thrun, S., Obermayer, K. (Eds) Advances in Neural Information Processing Systems 15: Proceedings of the 2002 Conference, MIT Press. ISBN 0-262-02550-7)

[2]. Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.

[3]. Yeh, I-Cheng, Yang, King-Jang, and Ting, Tao-Ming, "Knowledge discovery on RFM model using Bernoulli sequence, "Expert Systems with Applications, 2008

[4]. Rosenblatt, Frank. x. Principles of Neurodynamic: Perceptron's and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961

[5]. Rao, C.R. (1952) Advanced Statistical Methods in Multivariate Analysis, Wiley. (Section 9c)

[6]. Anderson, T.W. (1958) An Introduction to Multivariate Statistical Analysis, Wiley.

[7]. Binder, D.A. (1978) "Bayesian cluster analysis", Biometrika, 65, 31–38.

[8]. Binder, D.A. (1981) "Approximations to Bayesian clustering rules", Biometrika, 68, 275–285.

[9]. Har-Peled, S., Roth, D., Zimak, D. (2003) "Constraint Classification for Multiclass Classification and Ranking."

[10]. Evaluation of a Tree-based Pipeline Optimization Tool for Automating Data Science Randal S. Olson University of Pennsylvania olsonran@upenn.edu

Nathan Bartley University of Chicago bartleyn@uchicago.edu

Ryan J. Urbanowicz University of Pennsylvania ryanurb@upenn.edu

Jason H. Moore University of Pennsylvania jhmoore@upenn.edu

[This paper is based on this project]:

Determining the best classifier for predicting the value of a boolean field on a blood donor database using genetic algorithms. Papers With Code. (n.d.). https://paperswithcode.com/paper/determining-the-best-classifier-for