International Journal of Computer Applications (0975 - 8887)
Volume 176 - No.28, June 2020

# Statistical Approach for Predicting the Most Accurate Classification Algorithm for a Data Set in Analysis

Shriniwas Nayak
Savitribai Phule Pune University

Aditya Mahaddalkar
Savitribai Phule Pune University

## ABSTRACT

Classification algorithms under the category of data mining have widespread applications in the modern world finding their use in almost every field and area that aims at predicting an outcome class for some data instance. As a result of which many supervised classification algorithms have been studied in the field of machine learning. Many classification algorithms can be used to serve the purpose, K-Nearest Neighbor, Gaussian Naive Bayes, Decision Tree to name a few.

However even today it is a time consuming and complex task to decide the most suitable algorithm for the data under consideration. This article discusses an approach that predicts an algorithm that would produce best accuracy for the given data, depending upon internal data parameters : size of data, ratio of numerical attributes, count of outliers, average correlation, number of classes in target and average number of classes in attributes. This paper analyses the relation between the performance of K-Nearest Neighbor, Logistic Regression, Gaussian Naive Bayes and Decision Tree classification algorithms and internal data parameters thereby evaluating a generic approach to determine the most accurate algorithm and also studies some limitations, like the inability of incorporating external factors namely memory requirement and others.

## General Terms

Machine Learning, Statistical Analysis, Algorithm Prediction

## Keywords

Supervised Learning, Classification Algorithm, Decision Tree, Logistic Regression, K Nearest Neighbors, Naive Bayes

## 1. INTRODUCTION

Data mining is used for KDD (Knowledge Discovery in Database) that aims at finding some actionable insight from existing data. The main objective in classification tasks is to predict the outcome for a given data instance thereby helping to take corrective measures in the field of medical diagnosis, increase profits in case of stock market and general market analysis, reduce operational costs by concentrating on a particular section of probable customers in marketing and so on. The use and applications of these algorithms is widespread, the choice of a particular algorithm for data under consideration is to be made by a data scientist.

These classification algorithms fall under the category of supervised learning as these are provided with input tuples for which the output class is already known, depending on these tags the system learns to classify or tag a previously unseen data instance.

### 1.1 Motivation

The current procedure to decide the algorithm involves creating a subset of the data, which is assumed to be a succinct and complete representation of data, however this may not always be true. This data is used for training purposes and different algorithms are trained for this subset, then using some additional data from the actual data testing of these models is undertaken and then the most accurate algorithm is chosen [17]. This method at times may produce a sub optimal solution owing to the fact that it is difficult to find a true subset of the data that represents all different features, also the complexity involved in training many different models is high. This paper discusses an alternative approach to this problem wherein the system tries to identify similar data sets based on some internal parameters and predict the best classification algorithm for a particular data set.

### 1.2 Predictive approach in a nutshell

The approach primarily classifies a data set as categorical majority or numerical majority depending on the majority of attribute type present in the data set. For example the iris data created by Ronald Fisher will be classified as numeric as it has four attributes all of which are numerical in nature. Internal data parameters namely size of data, ratio of numerical attributes and number of classes in target are common to both the types are calculated before proceeding for specific data parameters depending type of data set categorical or numerical attribute heavy.

After calculation of the above mentioned generic data parameters and particular parameters depending upon the majority of categorical or numerical are calculated. Average correlation and normalised count of outliers for numerical data set, average of null hypothesis rejection and average of number of classes in attributes for categorical data set. Using the above-mentioned classification methods, the classification method with the maximum accuracy is added as the label for that data set; a neighbor based recommendation approach is used to identify which algorithm will perform the best for previously unseen similar data sets.

## 2. RELATED WORK

The selection of appropriate machine learning algorithms is dependent on data to be processed. The naive method of selecting the suitable algorithm proceeds by extracting and visualizing the attributes of data, dividing the data into training set and validation set and then training multiple machine learning models on the data [17]. The accuracy of these models are compared to determine the best model for data set under consideration.

This methodology works well for data sets with less data instances, data sets with less number of features, where machine learning models can run with good speed and provide results in a short span of time. Although, when large data sets come into picture, training each machine learning models on the data becomes time consuming and inefficient.

The article by S. Nikam [14] compares classification algorithms and helps to understand, that different classification algorithms are suitable for different data sets. N. Satyanarayana et al. [15] have studied the performance of classification algorithms with respected to data parameters like number of data instances, attributes in data, the noise-to-signal ratio in data and data extraneous factors such as accuracy comparison among different models, execution time required during training and prediction. Both the studies provide a qualitative analysis of classification methodologies.

F. Syeda et al. [19] provide a quantitative comparison of accuracy obtained by training classification algorithms on Iris data set, liver disorder data set and E-Coli data set. A similar analysis was carried out by R. Duriqi et al. [7] for Diabetes, Spam Base and Credit Approval data sets. The authors compare the algorithms under discussion with respect to parameters such as time required for execution, number of correct instances predicted, number of wrong instances predicted and respectively their accuracy on all the data sets. However, these approaches require every model to be trained on a data set and hence the methodology to determine the most accurate algorithm is cumbersome. A generic method that predicts the most accurate classification algorithm without training every model is required and this article tries to bridge this gap.

A naive approach is to train all machine learning models, visualize all the parameters of the data set such as plotting boxplots for detecting outliers, generating heatmaps for correlation matrix of the columns, plotting histograms of data attributes to observe the type of distribution and then selecting the most suitable machine learning model for all these parameters. This is a tedious process and requires many manual iterations [13], which we strive to automate and provide a good recommendation of appropriate machine learning model for any generic data set.

## 3. PREDICTIVE APPROACH BASED ON INTERNAL DATA PARAMETERS

Different classification algorithms prove to be better suited for data having different internal parameters, generic internal parameters considered are size of the data, fraction of numerical attributes in data and number of classes in the target variable. KNN functions well on data in which samples can have many class labels [3], Naive Bayes algorithm requires a very large number of data instances for better performance [12] and so forth.

Two different tables are generated, one for data sets which have numerical attributes and second for data sets that have categorical attributes in majority. Firstly for a particular data set under consideration the above mentioned generic internal data parameters are calculated and then depending on the type of the data set whether numerical/categorical attribute intensive, particular internal data parameters are calculated. All classification algorithms under consideration namely KNN, GNB, LogReg and DTree are used to train respective models and their accuracy is calculated and the algorithm with maximum accuracy is considered as the output for that data set. Then an entry is made in the respective table consisting of the generic, specific data parameters and the target which is an algorithm in this case.

### 3.1 Generic parameters

This section discusses the impact of the generic selected attributes on the performance of different classification algorithms and thereby analysing the need of assessing them to predict the best algorithm for an unknown data set.

#### *3.1.1 Size of data.*

Size of the data is the easiest to calculate amongst different parameters considered. Different algorithms prove to be suitable for different sizes of data sets. KNN algorithm proves to work well on small data sets, GNB on huge data sets and DTree on large data sets with respect to size [10]. This approach requires the data set to be devoid of any null values and erroneous entries. It is merely the total number of rows in the data set without the attribute titles.

#### *3.1.2 Ratio of numerical attributes.*

This attribute tries to indicate whether the data set has a majority of numerical attributes or categorical attributes. An attribute is classified as numerical if it has range based or interval scaled values and as categorical if it has nominal or ordinal values. Some algorithms perform better with categorical data while others with numerical data. Logistic Regression and Decision Tree are used when data sets contain mixed variable types [16]. In decision tree algorithm attributes are preferred to be categorical [5].

#### *3.1.3 Number of classes in target.*

Number of classes indicates the number of unique values in the target i.e. the number of labels present in the data set. Algorithms like KNN are better suited for multi-modal classification [3] and algorithms like logistic regression use one vs all strategy to classify data sets with more than two classes [4]. Hence it is considered in the analysis.

### 3.2 Numerical parameters

Numeric parameters are calculated only for those data sets that have a majority of numerical attributes. These attributes and the need for their analysis is discussed in this section.

#### *3.2.1 Average of Correlation Coefficient.*

Correlation helps to identify the dependence of values of one attribute with another. Value of Karl Pearsons product-moment coefficient of correlation lies between $+1$ and $-1$ where sign denotes the direction of relation and value the strength, for example, $1$ denotes complete dependence and $0$ complete independence. An upper triangular correlation matrix is formed and the average of

absolute values of this matrix is considered. GNB assumes that attributes are independent and hence this parameter is considered [4][8].

*3.2.2 Normalised Average of Outliers.*

Outlier is an observation that is distant from other observations and may affect the performance of algorithms for example if the value of hyper parameter k is small in the KNN algorithm then it is vulnerable to overfitting due to presence of noise [5]. Inter Quartile Range is calculated for each column and any data point that lies outside a specified range calculated using the the median and inter quartile range is considered as an outlier. The average of all attributes is divided by the size of the data set to obtain the final index. Algorithms like KNN and Naive Bayes are considered robust to noisy training data [12][10].

### 3.3 Categorical parameters

This section discusses the parameters which are considered only for categorical variables and their impact on classification algorithms.

*3.3.1 Average of null hypothesis rejection.*

Pearsons chi-squared statistical hypothesis test is used to determine the independence/dependence between categorical attributes. A null hypothesis is formulated that states H0 : The attributes are independent and alternative hypothesis HA : The attributes are dependent, using an appropriate level of significance and calculated degree of freedom either H0 is accepted or rejected. Ratio of number of cases where H0 is rejected to the total number of cases is considered. For n number of categorical columns $nX(n-1)/2$ tests are carried out. This attribute is considered as performance of algorithms depends on the correlation of attributes , for example Decision trees are not well suited for highly correlated data as over fitting may occur [16].

*3.3.2 Average of number of classes.*

This parameter finds the number of classes in each categorical attribute and takes the average of the same. For each attribute in the attribute list the number of unique values is the number of classes in that class, for example in attribute GENDER there are 3 classes namely male, female and prefer not say. It affects the number of splits at a decision node in the decision tree algorithm [9].

## 4. EXPERIMENTAL SETUP AND DATA SOURCES

### 4.1 Software Requirements

Python version 3.7.6 programming language and scikit-learn version 0.22.1 machine learning library were used for training and validating all the classification algorithms in discussion. scikit-learn library provides a plethora of supervised and unsupervised learning algorithms, from which four classification algorithms were selected as mentioned in earlier text. The data is of the format (n × p) where n is the number of instances and each instance consisting of p attributes, is provided to the library method fit of the respective classes of the algorithms to train the classification models. The predict method of the model predicts the classes from a given instance or multiple instances of the data. 5 fold cross validation technique was used, which is represented by cross_val_score function present in scikit-learn library. From the 5 respective accuracy scores, the mean accuracy score was assigned to the respective algorithm.

### 4.2 Data Sources

A total of 19 data sets having categorical attributes in majority and 21 data sets having numerical attribute in majority have been considered. All the considered data sets are open sourced and free for use and available on UCI Machine Learning Repository [2] and Kaggle [1]. The data sets have a lot of variety in instance counts, number of attributes, number of target classes as well as count of outliers and correlation factors.

## 5. ALGORITHM

The algorithms 1,2 and 3 receive the data set as input, for example iris.csv file. The generate metric algorithm (algorithm 1) calculates the generic parameters namely the size of the data, fraction of numerical attributes and number of classes in target and then depending on the value of fraction of numerical attributes appropriate algorithm is executed, if the value is greater than or equal to 0.5 then find_numerical_dataparameters and if less than 0.5 then find_categorical_dataparameters. The generate metric algorithm writes the generic parameters, specific parameters and the target classification algorithm in a table, two tables are maintained one for categorical attribute intensive data sets and other for categorical attribute intensive data sets.

For predicting the most accurate algorithm, the above algorithm is implemented to generate training data set, then without actually trying out different models on the test data set, depending on the type of the test instance the KNN algorithm is used on the corresponding training data set i.e. numerical or categorical generated by the algorithm.

**Algorithm 1** Generate training data set

```
1: procedure GENERATE_METRIC(data_set)
2:     remove missing or null values form data
3:     find accuracies for KNN, LogReg, GNB and DTree
4:     targetAlgorithm := classification algorithm with maximum accuracy
5:     size := number of rows in data set without attribute titles
6:     numericalAttributes := number of numerical attributes in data
7:     categoricalAttributes := number of categorical attributes in data
8:     RatioOfNumericalAttributes := numericalAttributes/(totalAttributes)
9:     numberOfClassesInTarget := number of unique values in target attribute
10:    if RatioOfNumericalAttributes ≥ 0.5 then
11:        find_numerical_dataparameters(data_set)
12:        add entry (size, averageOfCorrelationCoefficient, normalisedAverageOfOutliers, ratioOfNumericalAttributes, numberOfClassesInTarget, targetAlgorithm ) in numerical table
13:    else
14:        find_categorical_dataparameters(data_set)
15:        add entry (size, averageOfNullHypothesisRejection, averageOfNumberOfClassesInAttributes, ratioOfNumericalAttributes, numberOfClassesInTarget, targetAlgorithm ) in categorical table
```

**Algorithm 2** Find parameters for numeric major data sets

**procedure** FIND_NUMERICAL_DATAPARAMETERS($data\_set$)
  **while** listOfAttributes is not traversed **do**
    attribute := listOfAttributes.nextAttribute
    InterQuartile Range $IQR_{3,1}$ := $3^{rd}$ Quartile $Q_3$ - $1^{st}$ Quartile $Q_1$
    upperOutlier_limit := median $+1.5 \times IQR_{3,1}$
    lowerOutlierLimit := median $-1.5 \times IQR_{3,1}$
    $\forall$ value $\geq$ upperOutlierLimit OR value $\leq$ lowerOutlierLimit, Increment attributeOutlierCount
    totalOutlierCount := totalOutlierCount + attributeOutlierCount
  normalisedAverageOfOutliers := ((total outlier count) / number of attributes) / sizeOfData
  covarianceMatrix := find upper triangular covariance matrix for all attributes
  averageOfCorrelationCoefficient := (sum of absolute values in covarianceMatrix) / sizeOfMatrix

Algorithm 2 is used for calculating parameters specific to numerical attributes. The algorithms finds average frequency of outliers and creates a upper triangular correlation matrix and takes the average of the absolute values.

**Algorithm 3** Find parameters for categorical major data sets

**procedure** FIND_CATEGORICAL_DATAPARAMETERS($data\_set$)
  **while** listOfAttributes is not traversed **do**
    attribute := listOfAttributes.nextAttribute
    numberOfClassesInAttribute := number unique values for the attribute in data
    totalClasses := totalClasses + numberOfClassesInAttribute
  AverageClassesInAttributes := totalClasses / (numberOfAttributes)
  **while** unorderedPair from listOfAttributes not analysed **do**
    unorderedPair := listOfAttributes.nextPair
    $H_0$ : attributes are independent of each other
    $H_A$ : attributes are dependent on each other
    **if** $H_0$ rejected using chi square test **then**
      attributeChiOutcome := 1
    **else**
      attributeChiOutcome := 0
  averageOfNullHypothesisRejection := average of totalChiOutcome

Algorithm 3 is used for data sets with categorical attributes in majority. It calculates the average of frequency of classes in attributes. It calculates the ratio of null hypothesis rejection for all unordered pairs of attributes.

## 6. SUBSET OF GENERATED TRAINING DATA SET

The above presented algorithms were used for analysis, a variety of data sets were considered for the same. Table 1 and Table 2 represent a subset of the generated tables. Table 1, represents a subset of data sets where the count of numerical attributes exceeds that of categorical attributes. Table 2 represents a subset of data sets in which the count of categorical attributes exceeds that of numerical attributes.

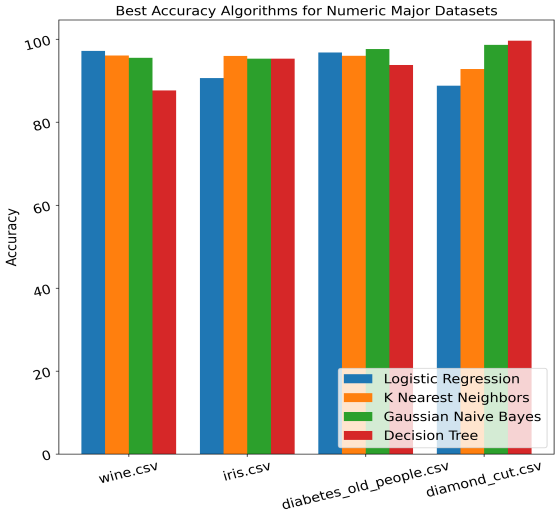


Fig. 1. Numeric major data set classification accuracy

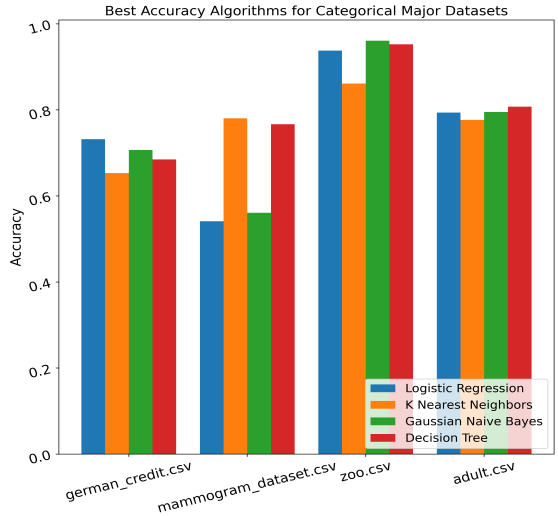


Fig. 2. Categorical major data set classification accuracy

Figure 1 and figure 2 represent the accuracy percentage for numerical and categorical majority data sets respectively, the intermediate results are obtained for the mentioned data sets for the four classification algorithms under consideration.

Depending on this intermediate result the target column of the training data set is populated, for example the maximum accuracy for iris data set is obtained using KNN algorithm and hence the target value for iris data set is KNN algorithm as is represented in table 1.

Table 1. Numerical Major Data sets (Subset of complete data)

| File Name | Size | Avg of correlation coeff | Normalised avg of outliers | Numeric Fraction | Class count | Target |
|---|---|---|---|---|---|---|
| wine.csv | 178 | 0.305 | 0.009 | 1 | 3 | Logistic Regression |
| iris.csv | 150 | 0.590 | 0.006 | 1 | 3 | KNN |
| diabetes_old.csv | 2818 | 0.244 | 0.005 | 0.875 | 4 | Gaussian NB |
| diamond_cut.csv | 599 | 0.139 | 0.016 | 0.75 | 4 | Decision Tree |
| spam.csv | 4601 | 0.061 | 0.117 | 1 | 2 | Logistic Regression |
| ecoli.csv | 336 | 0.294 | 0.0131 | 0.714 | 8 | KNN |
| ionosphere.csv | 351 | 0.238 | 0.068 | 0.941 | 2 | Decision Tree |
| transfusion.csv | 748 | 0.466 | 0.0324 | 1 | 2 | Logistic Regression |

Generated training data set for numerical major data sets, represents subset of complete data.

Table 2. Categorical Major Data sets (Subset of complete data)

| File Name | Size | Avg of null hypo rejection | Avg no of classes in attr | Numeric Fraction | Class count | Target |
|---|---|---|---|---|---|---|
| german_credit.csv | 1000 | 0.551 | 4 | 0.15 | 2 | Logistic Regression |
| mamogram.csv | 961 | 1 | 6 | 0.2 | 2 | KNN |
| zoo.csv | 101 | 0.516 | 2.25 | 0 | 7 | Gaussian NB |
| adult.csv | 32561 | 1 | 12.75 | 0.428 | 2 | Decision Tree |
| voting.csv | 435 | 0.967 | 0 | 3 | 2 | Logistic Regression |
| tic-tac-toe.csv | 958 | 0.667 | 0 | 3 | 2 | KNN |
| horse_colic.csv | 368 | 0.801 | 0.346 | 4.529 | 4 | Decision Tree |
| cmc_data.csv | 1473 | 0.857 | 0.222 | 3.143 | 3 | KNN |

Generated training data set for categorical major data sets, represents subset of complete data.

## 7. RECOMMENDATION SYSTEM BASED ON KNN

Based on the comparison model of recommendation system the K Nearest Neighbor algorithm was used for prediction of most accurate algorithm. The proposed system tries to identify data sets with similar internal parameters and the algorithm which works best for them, hence the KNN algorithm which finds the class for an unknown data instance depending upon similarity of parameters with neighbors [6] is most suited for this case.

### 7.1 Optimization of Hyper Parameters

To further increase the accuracy of the KNN model, the Bayesian Optimization technique for hyper parameter tuning has been used . Bayesian optimization strives to minimize an objective function for a set of parameters derived from a defined search space [18]. The search space of parameters, in this case, will be the hyper parameters for the K Nearest Neighbor model. These parameters include, number of neighbors which are compared with test point, the underlying algorithm used in the model as ball tree or k-dimensional tree, leaf size in ball tree or k-dimensional tree, the power parameter for the Minkowski distance p. Bayesian optimization techniques attempt to generate the next set of hyper parameters from the result of previous output of objective function so as to find the local minimum of the given objective function and thus, find the optimum set of hyper parameters for the KNN model.

## 8. RESULTS

Different classification algorithms were implemented on the generated data set in order to assess their accuracy. The results are represented in figure 3. As suggested by the graph, the KNN algorithm provides the maximum accuracy, this result provides further insight into the approach. The proposed approach relies on the hypothesis that particular classification algorithms provide better accuracy on data sets with suitable data parameters. The KNN algorithm predicts the class of a data instance depending on its neighbors, since for the generated data sets KNN algorithm provides maximum accuracy as compared to other algorithms, it can be learnt that performance of classification algorithm can be predicted for data sets with similar internal data parameters. Hence further analysis is presented using KNN algorithm.

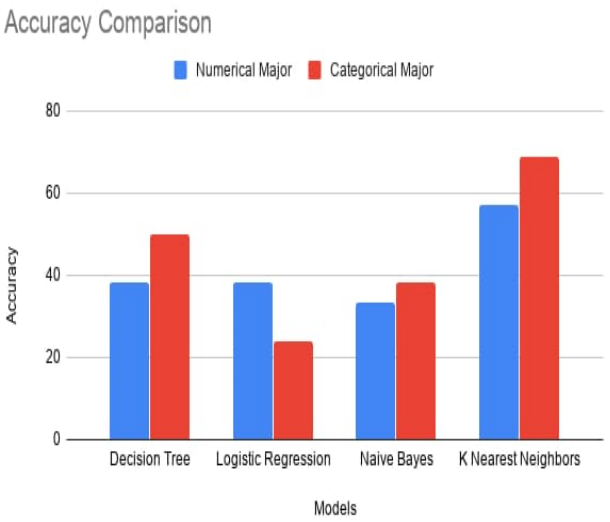


Fig. 3. Comparison of classification algorithms on generated data set

The KNN model model was trained for both the generated data sets, numerical as well as categorical, the subsets of which are represented in table 1 and 2. K-fold cross validation techniques are used to provide an accurate goodness of any machine learning model [11] on the given data set. The technique proceeds by dividing the data rows into K number of sets, training a model on K-1 sets and testing the model on the remaining set. This process

is repeated K times, each time the selection of training sets and testing sets are cycled through the data. 5-fold cross validation technique was used for generating accuracy score for the KNN based recommendation model for both data sets. The accuracy scores obtained after optimizing hyper parameters are represented in table 3.

Precision is defined as the ratio of correctly predicted positive observations to the total predicted positive observations. Recall is defined as the ratio of correctly predicted positive observations to the all observations in actual class. F1 score is the harmonic mean of precision score and recall score. Evaluation of the proposed approach with respect to these metrics is presented. The table 3 highlights the metrics precision score, recall score and F1 score of both categories of data sets namely numerical major and categorical major. To compute these scores, 5-fold cross validation technique was used and the weighted average of all the classes was considered while computing these metrics.

Table 3. Accuracy Table

| Table Type | Accuracy 1 | Accuracy 2 | Precision | Recall | F1 Score |
|---|---|---|---|---|---|
| Categorical | 83.33% | 68.33% | 0.6 | 0.6 | 0.6 |
| Numerical | 83.33% | 56% | 0.56 | 0.6 | 0.58 |

Accuracy 1 : Best Case Accuracy, Accuracy 2 : Cross Validation Accuracy

Table 4 represents a subset of the obtained result, the presented data sets have been used as test instances. The KNN algorithm predicts the most accurate algorithm for the test instance using the training data.

Table 4. Results (Subset of testing data set)

| Data set | Target algo | Predicted algo |
|---|---|---|
| adult.csv | DTree | DTree |
| zoo.csv | GNB | GNB |
| iris.csv | LogReg | KNN |
| wine.csv | LogReg | LogReg |

## 9. ADVANTAGES

This section discusses some advantages this approach provides over the current methodology used for selection of the most accurate algorithm for a given data set. Some advantages are as follows:

(1) The system saves the manual effort of analysing the data to choose the most suitable algorithm and the time required to train all possible classification models.
(2) This approach does not deal with the subset of the data but the complete data hence for choosing an algorithm the complete data contributes, there by eliminating the need of subset creation.
(3) The approach in the process of predicting the best classification algorithm performs a detailed statistical analysis if data which can help in understanding the data set under consideration.

## 10. LIMITATIONS

Certain limitations were also observed during the analysis of the proposed approach, they have been discussed below:

### 10.1 Impact of external factors

Some external requirements like memory and response time of a classification method are not incorporated in this method, further at certain times these may prove to be of higher significance than considering accuracy alone.

### 10.2 Building training data set

It can be difficult to build a comprehensive training data set for the model, each data instance in the training data set is a complete data set in itself, further it may be cumbersome to build such a data set from scratch as it requires each classification algorithm under analysis to be trained for every instance.

### 10.3 Data/Domain specific operations

Feature extraction, selection and engineering helps to improve the accuracy of algorithms and at times relative accuracy of different algorithms changes when trained for normal data set and preprocessed data set. However, it is extremely difficult to design a generic approach to feature engineering as it is highly domain and problem specific. The proposed approach cannot predict the accuracy for multiple algorithms after use of these preprocessing techniques.

## 11. CONCLUSION

From the observations made with respect to the data sets considered during the analysis, it can be concluded that a statistical approach can predict the most accurate algorithm for a data set with considerable accuracy.

It can be learnt from the results that, depending on some generic internal data parameters and some specific internal data parameters of data sets, it can be determined which classification algorithm is most likely to provide the best accuracy. The proposed approach eliminates the need of training every classification model under consideration for subset of the data set. The relative performance of classification algorithms can be predicted for data sets with similar internal data parameters.

However, it should also be noted that this approach serves as a generic indicator to a data scientist to choose an algorithm and the limitations of this approach should also be considered when put to practical use.

## 12. ACKNOWLEDGEMENT

The authors of this paper would like to extend their sincere gratitude towards Professor H.P. Channe (PICT), Professor Dr. S.S. Sonawane (PICT) and Professor B.D. Zope (PICT) for their invaluable suggestions and guidance.

The authors would also like to thank their family for their indispensable support and patience in the complete journey of this endeavour.

## 13. REFERENCES

[1] Kaggle, (accessed March 2020). https://www.kaggle.com/.
[2] University of California Irvine Machine Learning Repository, (accessed March 2020). https://archive.ics.uci.edu/ml/index.php.

[3] Hetal Bhavsar and Amit Ganatra. A comparative study of training algorithms for supervised machine learning. *International Journal of Soft Computing and Engineering (IJSCE)*, 2(4):2231–2307, 2012.

[4] Giuseppe Bonaccorso. *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[5] N. S. Chauhan. Decision tree algorithm explained, December 2019 (accessed March 2020). https://towardsdatascience.com/decision-tree-algorithm-explained-83beb6e78ef4.

[6] Thomas Cover and Peter Hart. Nearest neighbor pattern classification. *IEEE transactions on information theory*, 13(1):21–27, 1967.

[7] Rafet Duriqi, Vigan Raca, and Betim Cico. Comparative analysis of classification algorithms on three different datasets using weka. In *2016 5th Mediterranean Conference on Embedded Computing (MECO)*, pages 335–338. IEEE, 2016.

[8] Nir Friedman, Dan Geiger, and Moises Goldszmidt. Bayesian network classifiers. *Machine learning*, 29(2-3):131–163, 1997.

[9] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[10] Sayali D Jadhav and HP Channe. Comparative study of k-nn, naive bayes and decision tree classification techniques. *International Journal of Science and Research (IJSR)*, 5(1):1842–1845, 2016.

[11] Ron Kohavi et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Ijcai*, volume 14, pages 1137–1145. Montreal, Canada, 1995.

[12] V Krishnaiah, G Narsimha, and N Subhash Chandra. Survey of classification techniques in data mining. *International Journal of Computer Sciences and Engineering*, 2(9):65–74, 2014.

[13] Gang Luo. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics*, 5(1):18, 2016.

[14] Sagar S Nikam. A comparative study of classification techniques in data mining algorithms. *Oriental journal of computer science & technology*, 8(1):13–19, 2015.

[15] N Satyanarayana, CH Ramalingaswamy, and Y Ramadevi. Survey of classification techniques in data mining. *International Journal of Innovative Science, Engineering & Technology*, 1:268–278, 2014.

[16] Emc Education Services. Data science and big data analytics: Discovering, analyzing, visualizing and presenting data. pages 205–229, 2015.

[17] R. Shaikh. Choosing the best algorithm for your classification model, November 2018 (accessed March 2020). https://medium.com/datadriveninvestor/choosing-the-best-algorithm-for-your-classification-model-7c632c78f38f.

[18] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[19] Farha Syeda, Mustafa Ali Baig Mirza, Ali Baig, and M Pawar. Performance evaluation of different data mining classification algorithm and predictive analysis. *IOSR*, 10, 01 2013.