Analysis of Classification Algorithms on Big Data

Poonam[1] , Aditi Mittal[2]
[1]Assistant Professor, Department of Computer Science , Arya PG College, Panipat
[2]Assistant Professor, Department of Computer Science , Arya PG College, Panipat

ABSTRACT

With the advent of IoT, cloud computing and other latest technologies, a huge amount of data is generated every second of time from multiple sources. It is a real challenge to handle, store and analyze this data using conventional DBMS thus giving birth to Big data technology. In this, we need to classify the data into sets. Many classification algorithms classify the data into classes or categories. Classification algorithms aim at eliminating redundant, or irrelevant features that may deteriorate the classification performance. However, traditional methods lack enough scalability to cope with datasets of millions of instances and extract successful results in a delimited time. Classification techniques used to large transactional databases make it easier for users to get the information they need from massive datasets. Unsupervised and supervised classification are the two basic classification techniques. The goal of this work was to investigate several supervised classification algorithms. In this paper , we have taken a data set of credit card fraud usage from kaggle on which we performed classification algorithms such as Decision tree, Random forest, Logistic Regression and Support Vector Machine on the data set and analyzed the results. We found the best accuracy by Random Forest Algorithm when the test size is 0.3

KEYWORDS*:* Big data, Classification Algorithms, Big Data Analysis etc.

INTRODUCTION:

Big data means data which can not be handled by traditional methods such as RDBMS. RDBMS can only hold and manage the data in tabular form. Nowadays, with IoT and sensors , it is not feasible that all the data is

ISBN: 978-81-955611-3-1

arranged in tabular form. We have not only structured data, but we have unstructured data and semi structured data too. This kind of data can not be stored ,handled and analyzed by traditional techniques. Big data faces three major problems which are data accessing and computing, data Privacy and domain knowledge and the last challenge is data mining algorithms. Big data Mining Algorithms are designed to address the challenges posed by large amounts of data, scattered data distribution, and complex and dynamic properties. Big data mining algorithms have three major stages shown in figure 1.

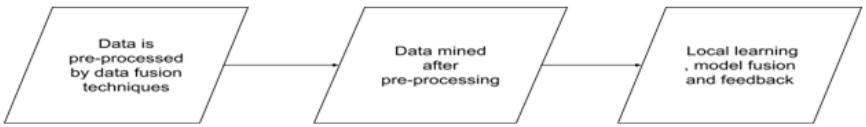


Figure 1: Big Data Mining Algorithm stages

Classification techniques are used to solve the above challenges which classify the big data according to the format of the data that must be processed, the type of analysis to be applied, the processing techniques at work, and the data sources for the data that the target system is required to acquire, load, process, analyze and store. Classification technique is an essential component of machine learning as well as big data. In this, data sets are analyzed and scrutinized for the patterns. Then specific algorithms are then designed according to the dataset for the pattern recognition. Pattern classification algorithms can be classified in multiple ways

- Supervised Learning
- Unsupervised learning
- Semi supervised learning.

In this paper, we have taken a data set of credit fraud from kaggle and performed various classification algorithms and found their results. We performed supervised algorithms such as Decision Tree classifier, Support Vector Machine, Random forest Classifier and Logistic Regression in this paper.

CLASSIFICATION ALGORITHMS :

1) **Decision Tree classifier :** It is a supervised learning algorithm in which we classify all data sets in separate classes by classifying them

ISBN: 978-81-955611-3-1

according to data features. Whenever we ask a question that results in either yes or no , we get a new data set. The process is repeated until all the data sets are in isolated classes and it can not be further divided or there is no need for more classification.

2) **Support Vector Machine :** SVM algo is also a supervised learning algorithm. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called support vectors, and hence the algorithm is termed as Support Vector Machine.

3) **Random forest classifier Algorithm :** It is a classification algorithm that uses a number of datasets according to various dataset features and then it uses the average for the final classification. The more the number of datasets , the higher the accuracy. It is considered to be the most accurate algorithm among all supervised learning algorithms.

4) **Logistic Regression:** The goal of logistic regression is to identify the best fit between a dependent variable and a set of independent variables. In this algorithm we predict the categorical dependent variable. The output values are not 0 or 1 but in the range of 0 and 1.

CLASSIFICATION ANALYSIS:

For the research , we have taken a credit card fraud dataset from kaggle. We applied a classification algorithm on this data. This data contains 31 columns and 284807 rows. In this data 30 columns are independent and 1 column named class is dependent. Class field contains 2 values either 1 or 0. 0 represents fraud and 1 represents fraud. If we performed a group by query on a data set by grouping the Class field , the result is shown in figure 2

Data from " class" column	Rows
0	284315
1	492

```
import seaborn as sns sns.countplot(card['Class'],label="Credit card Fraud")
```

ISBN: 978-81-955611-3-1

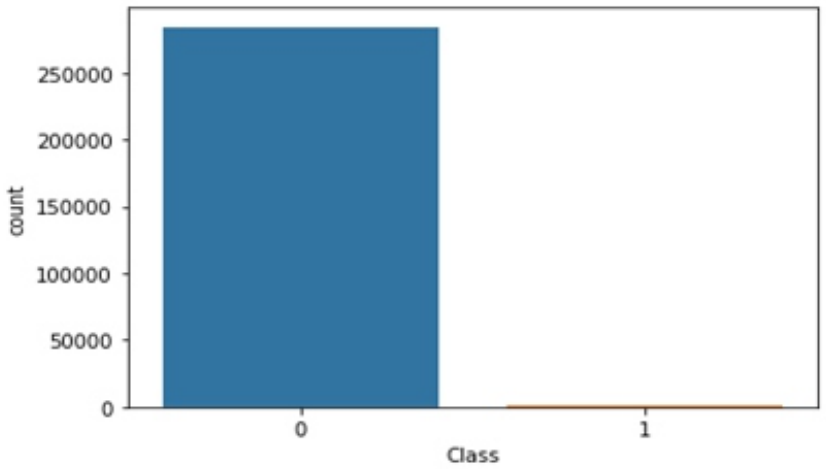


Figure 2: Fraud or non fraud data of data set

```
Data contains
[284807 rows x 31 columns] dimensions of
the date {} (284807, 31)
```

We used scikit learn model selection for splitting the data into test data and train data.

```
from sklearn.model_selection import train_test_split
train_data,test_data,train_label,test_label=train_test_split(card.loc[:,card.columns!='Class']
,card['Class'],stratify=card['Class'],random_state=100,test_size=0.3) len(train_data)
```

Table 1: Comparison between different Classification Algorithm with test size=0.3

Classification Algorithm	Testing Accuracy	Training Accuracy	Executiontime in Seconds
Decision Tree Classifier	0.9990754069964772	1.0	14.243760108947754
SVM	0.9982678510820079	0.9982745129511847	32.006568908691406
Random Forest Classifier	0.9994850368081645	1.0	180.09995317459106
Logistic Regression	0.9989466661985184	0.9990168736582332	5.376386642456055

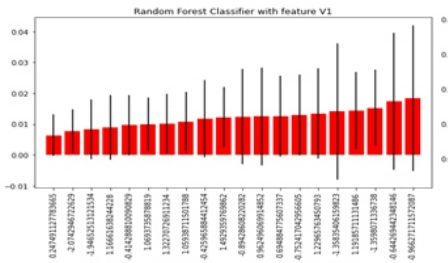
ISBN: 978-81-955611-3-1

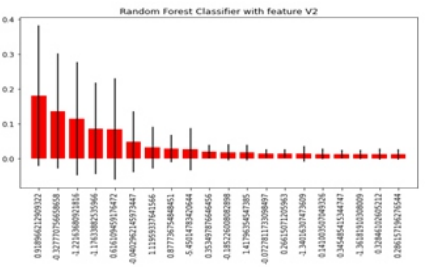
```
from sklearn.model_selection import train_test_split
```

```
train_data,test_data,train_label,test_label=train_test_split(card.loc[:,card.colu
mns!='Class'],card['Class'],stratify=card['Class'],random_state=100,test_size= 0.5)
len(train_data)
```

Table 2: Comparison between different Classification Algorithm with test size=0.5

Classification Algorithm	Testing Accuracy	Training Accuracy	Execution Time in Seconds
Decision Tree Classifier	0.9990519929215471	1.0	12.49266409 8739624
SVM	0.998272520434819 3	0.9982725083 038981	30.03260612 487793
Random Forest Classifier	0.9995084407741356	1.0	116.732189893 72253
Logistic Regression	0.9990941265694784	0.9991924327436 922	4.23209691047 6685





ISBN: 978-81-955611-3-1

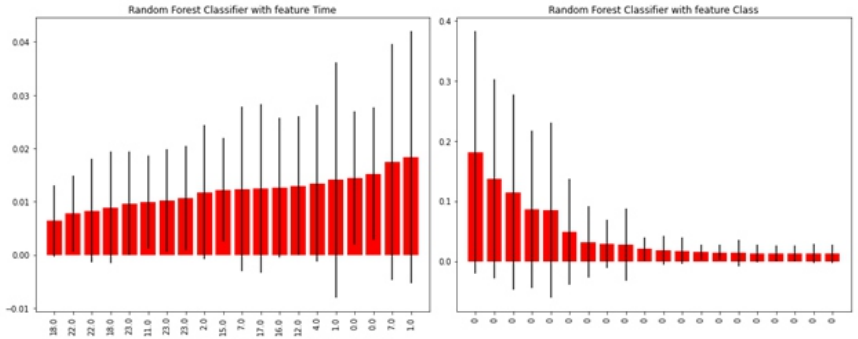


Figure 3: Result of Random Forest Classifier with Feature V1,V2,Time and Class

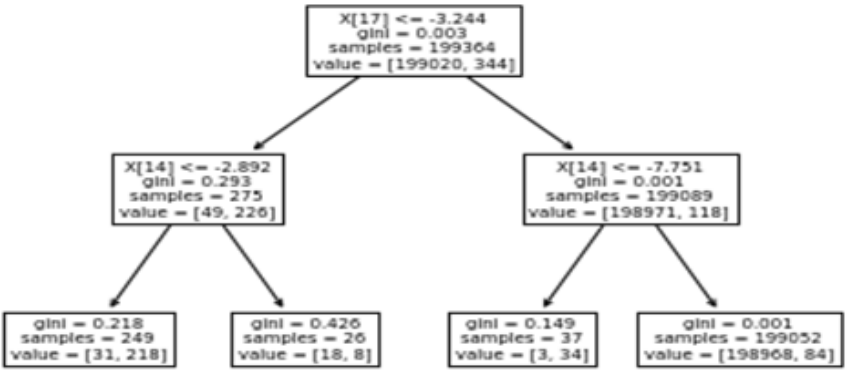


Figure 4: Result of Decision Tree Classifier with Max Depth=2

CONCLUSION

Every classification algorithm has its merits as well as demerits and all of them can be used effectively in Big Data with the different uses and different accuracies. The decision tree algorithm is the simplest to implement in all the algorithms whereas random forest technique works on multiple trees with different features and the output is the average of all the trees. Random forest produces the most accurate result in most cases. SVM works better when dealing with continuous features. Logistic regression execution time is the least of all the classification algorithms. The paper presents results of decision tree as well as random forest techniques with max depth 2 .

ISBN: 978-81-955611-3-1

REFERENCES

1. Dataset: https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-datasets/d ata
2. Kotsiantis, Sotiris B., Ioannis Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification techniques." Emerging artificial intelligence applications in computer engineering 160.1 (2007): 3-24.
3. G. Kesavaraj, Dr. S. Sukumaran, "A Study on Classification Techniques in Data Mining," 4th ICCCNT – Tiruchengode, India, 31661, July 4 - 6, 2013, IEEE
4. Dietterich, T. G. (1998), Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation, 10(7) 1895–1924
5. Çatak, F. Ö., & Balaban, M. E. (2016). A MapReduce-based distributed SVM algorithm for binary classification. Turkish Journal of Electrical Engineering & Computer Sciences, 24(3), 863-873.
6. Szczerbicki, Edward. "Management of complexity and information flow." Agile Manufacturing: The 21st Century Competitive Strategy, 1st ed. London: Elsevier Ltd (2001): 247-63.
7. Osisanwo, F. Y., et al. "Supervised machine learning algorithms: classification and comparison." International Journal of Computer Trends and Technology (IJCTT) 48.3 (2017): 128-138.
8. Padillo, Francisco, José María Luna, and Sebastián Ventura. "Evaluating associative classification algorithms for Big Data." Big Data Analytics 4.1 (2019): 1-27.
9. Nayak, Sinkon, et al. "Comparative analysis of heart disease classification algorithms using big data analytical tool." International Conference on Computer Networks and Inventive Communication Technologies. Springer, Cham, 2019.
10. Fahad, Adil, et al. "A survey of clustering algorithms for big data: Taxonomy and empirical analysis." IEEE transactions on emerging topics in computing 2.3 (2014): 267-279.

ISBN: 978-81-955611-3-1