International Journal of Scientific Research & Engineering Trends
Volume 7, Issue 4, July-Aug-2021, ISSN (Online): 2395-566X

Glass Identification Using Extreme Gradient Boosting Algorithm

Pulkit Dhingra
Department of Computer Science,
Babu Banarsi Das Northern India Institute of Technology
Lucknow, Uttar Pradesh, India.
1805610079@bbdniit.ac.in

***Abstract*- The Discovery of important information from criminal evidence is essential to make an effective criminological investigation. It is necessary to analyze this data of evidence and get answers that will help in the study. Machine learning techniques provide a bundle of state-of-the-art algorithms to analyze the data to get results efficiently. Extreme Gradient Boosting (XG-Boost) is one of the most successful machine learning algorithms used in classification problems. This paper applies the XG-Boost technique to help criminological investigators identifying the type of glass gathered as evidence while undergoing an investigation. The method can also be beneficial for other use cases that include the segregation of glass based on the materials incorporated in it.**

***Keywords*- Machine Learning, XG-Boost, Glass Classification.**

I. INTRODUCTION

The study of the glass classification problem was motivated by criminological investigation. At the crime scene, the glass left can be used as evidence when correctly identified. A frequent use-case is comparing glass from a crime scene with glass particles associated with a suspect. Such glass particles are often exceedingly small. It is essential to identify and compare these small glass fragments that may be significant in a forensic context [1][2][12].

The quantitative analysis of glass gives the oxide concentration for up to 15 elements. In most cases, oxides of Sodium, Magnesium, Aluminum, Silicon, Potassium, and Calcium occur at levels high enough for measurement. [1] The main aim of the paper is the classification of glass fragments based on the components they constitute. The algorithm proposed for the glass classification is the Extreme Gradient Boost classification method. It is one of the most commonly used data mining techniques in pattern recognition and classification problems.

The dataset used in this paper is the Glass Identification Dataset taken from the UCI Machine Learning Repository [3]. Jupyter Notebook, an open-source software, was used throughout this study as a tool for data modeling and analysis using the Python Programming Language.

The remaining paper deals with the following sections. Section II deals with related works in the field of glass classification. Section III describes the glass identification dataset. The fourth section deals with the methodology of data analysis. The data analysis results and conclusion are shown consequently in sections V and VI.

II. RELATED WORK

There had been few pieces of research conducted related to glass classification. All of them have been applying various Machine learning techniques that include fuzzy clustering [6] and many variants of the KNN technique such as feature weighting [4], AdaBoost [5], locally adaptive KNN [7], bagging, kernel density, voted KNN [12] and support vector machine over same glass identification dataset. The results comparison with this paper is shown based on the classification accuracy in table I. They are all lower than the results achieved in this paper.

Table 1. A Classification Accuracy in Glss Dataset in Related Works.

Serial Number	Method Detail	
	Method	Accuracy Rate
1	Boosting NN [5]	75.6%
2	Naïve KNN [5]	73.2%
3	Adaptive metric NN [7]	75.2%
4	Discriminant Adaptive NN [7]	72.9%
5	C4.5 Decision Tree[7]	68.2%
6	Wilson Editing [8]	67.4%
7	KNN based Voting[12]	80.3%

The most recent research work was by Mashael S. Aldayel of the Department of Information Technology King Saud University Saud Arabia, Riyadh [12].

The paper proposed voting based on K Nearest Neighbour Technique that involves testing the data on various K values to get the optimum value. The model was able to achieve an accuracy of 90.09 percent.

III. DATASET DESCRIPTION

The UCI Machine Learning Repository provides a variety of datasets. One of them is the glass identification dataset [3] which determines the type of glass based on its components. There are 214 data points in the dataset and ten attributes or features shown in table II. The attributes are measured using weight percent in their corresponding oxide.

- Id number: 1 to 214
- RI: refractive index
- Na: Sodium
- Mg: Magnesium
- Al: Aluminum
- Si: Silicon
- K: Potassium
- Ca: Calcium
- Ba: Barium
- Fe: Iron

There are seven categories of glass materials.

- 1.-> building_windows_float_processed
- 2-> building_windows_non_float_processed
- 3-> vehicle_windows_float_processed
- 4-> vehicle_windows_non_float_processed (none in this database)
- 5-> containers
- 6-> tableware
- 7-> headlamps

IV. METHODOLOGY

The proposed method to classify glass is the Extreme Gradient Boosting technique. This research focuses on using the tuned XG-boost model to enhance the accuracy of glass classification. The practical aspects of the study use python as the programming language.

This study goes through two phases; data preparation and data modeling. The data preparation phase intends to prepare the dataset for the second phase. The second phase includes using an XG-Boost classifier to construct a highly accurate prediction model for the glass identification problem.

1. Data Preparation:

The dataset is present in a .data extension format and contains 214 data points. Initially, the dataset has to be converted to a CSV format to create a dataframe for effective processing. We subdivided the data into two sets to representing the test and train dataset in the ratio of 85 to 15, where 85 percent of datapoints comprises the training data, and the remaining fifteen percent the test data.

1.1 **Training Data:** The training data constitute the data points on which the model training will take place. While evaluating these data points will help to understand the model's performance over the data it had already seen.

1.2 **Test Data:** The test data constitute the data points that are new for the model. It will help in understanding the model's performance over unseen use cases. The data was already in a clean state with no missing values. The features we also not correlated. Since we intend to use a tree-based model, we need not use any feature scaling or normalization method.[12]

2. Data Modeling:

We fed the training data into the XG Boost classifier. The XG Boost classifier is a popular method in a wide range of classification problems. It is efficient for both small and large datasets. [9] XG Boost is a boosted ensemble technique. It means the knowledge from residuals of the earlier decision tree is forwarded to enhance the new tree.

XG-Boost is an extension of the Decision Tree algorithm that involves building up a tree-based structure from the features. It is built based on entropy and information gain. This tree enhanced after multiple iterations of each data point using the residual loss to get a Gradient Boosted tree at the extreme. Hyperparameter tuning while training the model enhances its accuracy. Tuning the hyperparameters may lead to a rapid increase in the performance of the model. This paper deals with tuning hyperparameters of the XG-Boost algorithm.

The parameters tuned are

2.1 **colsample_bylevel:-** It defines the subsample ratio of columns for each level. Subsampling occurs once for every new depth level reached in a tree. [8]

2.2 **colsample_bytree:-** It is the subsample ratio of columns when constructing each tree. [8]

2.3 **Gamma:-** It is the minimum loss reduction required to make a further partition on a leaf node of the tree. The larger the gamma value is, the more conservative the algorithm will be.[8]

2.4 **max_depth:-** It is the maximum depth of a tree. Increasing this value will make the model more complex and more likely to overfit.

2.5 **min_child_weight:-** It is the minimum sum of instance weight (hessian) needed in a child.[8]

2.6 **nthread:-** Number of parallel threads used to run XGBoost.

2.7 **n_estimators:-** It is the number of trees (or rounds) in an XGBoost model.

Though the XG-Boost algorithm is adequate for most of the use cases, it has few drawbacks. The algorithm has lots of hyperparameters for fine-tuning and, it may lead to overfitting in some cases. Thus it is essential to test the results while making changes in each of the parameters

V. RESULT ANALYSIS

The study shows that the XG-Boost models perform better than other conventional ways used for glass classification. As a result, identifying the glass type using the XG-Boost classifiers is accurate, with an accuracy rate of 90.90 percent over the test data. I firstly pre-process the glass dataset. Then divided it into parts (train and test set) for further analysis of the model. Training of the model took place using the training dataset. By using various performance metrics, we validated the model performance.

This include,

- **Precision Score:** Model precision score represents the ability of the model to correctly predict the positives out of all the correct predictions it made[10].
- **Recall Score:** Model recall score represents its ability to correctly predict the positives out of actual positives from the data. [10]
- **Accuracy Score:** Model accuracy represents the ability to make correct predictions.[10]
- **F1 Score:** Model recall score represents the harmonic mean of precision and recall score.[10]
- **Confusion Matrix:** It is a table representing the performance of a classification model on a set of data whose actual values are known. [10]

We tested the model on both training and testing data to test the model performance on both seen and unseen data.

Table 2. Accuracy Score of Model.

Accuracy Score	
Training Data | Seen data	98%
Testing Data | Unseen data	90.90%

Table 3. Precision Score of Model.

Precision Score	
Training Data | Seen data	97.8%
Testing Data | Unseen data	90.9%

Table 4. Recall Score of Model.

Recall Score	
Training Data | Seen data	97.8%
Testing Data | Unseen data	90.9%

Table 5. F1 Score of Model.

F1 Score	
Training Data | Seen data	97.8%
Testing Data | Unseen data	90.9%

Another accuracy measurement is the confusion matrix in Tables VI and VII. Each element in the confusion matrix is a count of instances. Rows in the matrix represent the actual class of the instances, and columns represent the predicted class. [12]

For training data, the matrix shows the following result:-

- Fifty-Nine cases of class 1 were correctly classified and, two incorrectly classified of class 2.
- Sixty-six data points of class 2 were correctly classified and, there was no incorrect classification in others.
- Fifteen cases of class 3 were correctly classified and, there was no incorrect classification in others.
- Ten data points of class 5 were correctly classified and, there was no incorrect classification in others.
- Six cases of class 6 were correctly classified and, there was no incorrect classification in others.
- Twenty-Two data points of class 7 were correctly classified and one incorrectly labeled of class 1.

For testing data, the matrix shows the following result:-

- Nine cases of class 1 were correctly classified and, one incorrectly labeled of class 2.
- Ten data points of class 2 were correctly classified and, one incorrectly labeled of class 1.
- Two cases of class 3 were correctly classified and, there was no incorrect classification in others.
- Three data points of class 5 were correctly classified and, one incorrectly labeled of class 2.
- Three cases of class 6 were correctly classified and, there was no incorrect classification in others.
- Six data points of class 7 were correctly classified and, there was no incorrect classification in others.

Table 6. Confusion Matrix Training Data.

Predicted Class							
1	2	3	4	5	6	7	
59	2	0	0	0	0	0	1
0	66	0	0	0	0	0	2
0	0	15	0	0	0	0	3
0	0	0	0	0	0	0	4
0	0	0	0	10	0	0	5
0	0	0	0	0	6	0	6
0	0	0	0	1	0	21	7

Table 7. Confusion Matrix Testing Data.

Predicted Class							
1	2	3	4	5	6	7	
8	1	0	0	0	0	0	1
1	9	0	0	0	0	0	2
0	0	2	0	0	0	0	3
0	0	0	0	0	0	0	4
0	1	0	0	2	0	0	5
0	0	0	0	0	3	0	6
0	0	0	0	0	0	6	7

VI. CONCLUSION

The analysis of the result shows the effectiveness of the XG-Boost model over the other conventional techniques used for glass classification. Fine-tuning the model further

enhances the accuracy to 90.90%. It indicates that the model will be accurate for 90.90 percent of its predictions. Classification of glass is helpful in many use cases. Classifying various types of glass material can be evidence on a criminal investigation or used in glass segregation before recycling.Furthermore, this paper has raised some interesting possibilities for further research with tuning XG-Boost classifiers for different datasets.

VII. CODE

- https://colab.research.google.com/drie/1zb7tkSlAW7qg KV XzinLKgQfHUnu71hlG?usp=sharing
- https://drive.google.com/drive/flders/1hlwmW1Q9I7D0i R bGFgHmctHD1D1OosWN?usp=sharing
- https://codeocean.com/capsule/69272/tree

ACKNOWLEDGMENT

The author would like to thank the partial support funded by the Department of Computer Science at Babu Banarsi Das Northern India Institute of Technology.

REFERENCES

[1] K. W. Terry, A. Van Riessen, and B. F. Lynch, Identification of Small Glass Fragments for Forensic Purposes. Government Chemical Laboratories, Criminology Research Council (Australia) and Western Australian Institute of Technology, 1983.
[2] WEKA, Waikato Environment for Knowledge Analysis Version 3.6.3, New Zealand, 1999-2010, online accessed: [http://www.cs.waikato.ac.nz /~ml/WEKA].
[3] German, B. was also misspelled as erman, (2012). UCI Machine Learning Repository Glass Identification Dataset, September 1987, online accessed: [http://archive.ics.uci.edu/ml/datasets/Glass+Identification].
[4] D. P. Vivencio, E. Hruschka, M. Nicoletti, E. dos Santos, and S. Galvao, Feature-weighted k-Nearest Neighbor Classifier, Foundations of Computational Intelligence, 2007. FOCI 2007. IEEE Symposium, pp. 481–486, 2007.
[5] V. Athitsos and S. Sclaroff, Boosting Nearest Neighbor Classifiers for Multiclass Recognition, Computer Vision and Pattern Recognition Workshops, 2005. CVPR Workshops. IEEE Computer Society Conference, pp. 45–45, 2005.
[6] S. Ruying and H. Rongcang, Data Mining Based on Fuzzy Rough Set Theory and its Application in the Glass Identification, Information, and Automation, 2009. ICIA'09. International Conference on, pp. 154–157, 2009.
[7] C. Dorneniconi, J. Peng, and D. Gunopulos, An Adaptive Metric Machine for Pattern Classification, Advances in Neural Information Processing Systems 13, vol. 13, p. 458, 2001.
[8] https://xgboost.readthedcs.io/en/latest/parameter.html
[9] https://www.quora.com/What-are-the-advantages-and-disadvantages-of-XGboost-I-know-they-are-very-good-for-supervised-learning-but- what-are-there-limitations-Do-they-still-perform-well-on-noisy-data
[10] https://machinelearningmastery.com/precision-recall-and-f-measure-f or-imbalanced-classification/
[11] Tianqi Chen, Carlos Guestrin, XGBoost: A Scalable Tree Boosting System, 10 June 2016.
[12] Mashael S. Aldayel, K-Nearest Neighbor Classification for Glass Identification Problem, Department of Information Technology, King Saud University, Saud Arabia, Riyadh, 2013