2022 IEEE NIGERCON

2022 IEEE Nigeria 4th International Conference on Disruptive Technologies for Sustainable Development (NIGERCON) | 978-1-6654-7978-3/22/$31.00 ©2022 IEEE | DOI: 10.1109/NIGERCON54645.2022.9803172

Machine Learning Models for Predicting Bank Loan Eligibility

Ugochukwu .E. Orji
Dept. of Computer Science
University of Nigeria, Nsukka
Enugu, Nigeria
ugochukwu.orji.pg00609@unn.edu.ng

Chikodili .H. Ugwuishiwu
Dept. of Computer Science
University of Nigeria, Nsukka
Enugu, Nigeria
chikodili.ugwuishiwu@unn.edu.ng

Joseph. C. N. Nguemaleu
Dept. of Computer Science
University of Nigeria, Nsukka
Enugu, Nigeria
nguemaleu.ngako.dp000058@unn.edu.ng

Peace. N. Ugwuanyi
Dept. of Computer Science
University of Nigeria, Nsukka
Enugu, Nigeria.
Peace.ugwuanyi@unn.edu.ng

***Abstract* — Machine learning algorithms are revolutionizing processes in all fields including; real-estate, security, bioinformatics, and the financial industry. The loan approval process is one of the most tedious task in the banking industry. Modern technology such as machine learning models can improve the speed, efficacy, and accuracy of loan approval processes. This paper presents six (6) machine learning algorithms (Random Forest, Gradient Boost, Decision Tree, Support Vector Machine, K-Nearest Neighbor, and Logistic Regression) for predicting loan eligibility. The models were trained on the historical dataset 'Loan Eligible Dataset,' available on Kaggle and licensed under Database Contents License (DbCL) v1.0. The dataset was processed and analyzed using Python programming libraries on Kaggle's Jupyter Notebook cloud environment. Our research result showed high-performance accuracy, with the Random forest algorithm having the highest score of 95.55% and Logistic regression with the lowest score of 80%. Our Models outperformed two of the three loan prediction models found in the literature in terms of precision-recall and accuracy.**

Keywords— KNN, SVM, Bagging and Boosting techniques, Efficient ML Algorithms, Loan approval prediction.

I. INTRODUCTION

Like many other business ventures, the banking sector is increasingly looking to take advantage of the opportunities presented by modern technologies to improve their processes, productivity and reduce costs. According to [1], the predictive analytics feature of Machine learning was the most utilized feature for applications in the banking sector worldwide in 2020. The success or failure of most lending platforms largely depends on their ability to evaluate credit risk [2]. The loan approval process is a challenging task for any financial institution. Before giving credit loans to borrowers, the bank decides whether the borrower is bad (defaulter) or good (non-defaulter). This paper focused on developing Machine Learning (ML) models to predict loan eligibility, which is vital in accelerating the decision-making process and determining if an applicant gets a loan or not. Our objectives in this study include; (1) Clean and Preprocess the data for modeling, (2) Perform Exploratory Data Analysis (EDA) on the dataset, (3) Build various ML models to predict loan eligibility, and (4) Evaluate and Compare the different Models built.

II. REVIEW OF RELATED LITERATURE

The authors in [3] carried out a systematic literature review to identify and compare the best fit ML-based models for credit risk assessment. The authors aimed to show the various ML algorithms utilized by researchers for credit assessment of rural borrowers, especially those with inadequate loan history. Their finding showed that the ML algorithms we utilized in this research were widely used and showed great results.

The adverse impact of low loan repayment rates on banks is a major issue globally, and banks are looking for more effective ways to handle loan approval processes. The authors in [4] evaluated the loan default prediction of the Chinese peer-to-peer (P2P) market using R.F, XGBoost, GBM, and Neural Network machine learning models. Their four models exceeded 90% accuracy, with RF being the superior model. This research is closely related to our study in terms of methods used and algorithms deployed; however, they aimed to predict P2P loan default while we aimed to predict customers' eligibility for loans.

In their research, [5] deployed various ensemble ML techniques such as AdaBoost, LogitBoost, Bagging, and Random Forest model to predict loan approval of bank direct marketing data. Their research result showed that AdaBoost had the highest accuracy of 83.97%. When compared to our study, the SMOTE technique we utilized to balance our dataset proved to be the key difference as our models achieved better performance.

The research by [6] studied actual bank credit data to predict customers' creditworthiness and help the banks formulate an automated risk assessment system. They deployed different ML algorithms, including; neural network, naive Bayes, KNN, decision tree, and ensemble learning algorithms. Their model accuracy ranged from 80% to 76% respectively, which is also below the accuracy of our models.

III. METHODOLOGY

This research was done using Python on Kaggle's Jupyter Notebook cloud environment. The proposed model predicts customers' loan eligibility based on the available data. The input to the model includes attributes from the dataset, as shown in table 1. The output from the model is a decision on whether the customer is eligible to get the loan. The following section discusses the dataset and explains the methods used to cleanse and preprocess the dataset for modeling.

978-1-6654-7978-3/22/$31.00 ©2022 IEEE

A. Dataset

The dataset used in this study is the historical dataset 'Loan Eligible Dataset,' available on Kaggle [7] and licensed under Database Contents License (DbCL) v1.0. Table 1 below gives a brief description of the dataset attributes.

Table 1: Dataset description

Variable Name	Description	Data Type
Loan_ID	Loan reference number (Unique I.D.)	Numeric
Gender	Applicant gender	Categorical
Married	Applicant marital status	Categorical
Dependents	Number of family members	Numeric
Education	Applicant educational qualification (graduate or not graduate)	Categorical
Self_Employed	Applicant employment status (yes for self-employed, no for employed/others)	Categorical
Applicant_Income	Applicant's monthly salary/income	Numeric
Coapplicant_Income	Additional applicant's monthly salary/income	Numeric
Loan_Amount	Loan amount	Numeric
Loan_Amount_Term	The loan's repayment period (in days)	Numeric
Credit_History	Records of applicant's credit history (0: bad credit history, 1: good credit history)	Numeric
Property_Area	The location of the applicant's home (Rural/Semi-urban/Urban)	Categorical
Loan_Status	Status of loan (Y: accepted, N: not accepted)	Categorical

B. Data preprocessing and Analysis

To ensure optimal performance of the model, the following techniques were deployed to analyze and preprocess the data for modeling;

1. Synthetic Minority Oversampling Technique (SMOTE): This technique is highly effective for handling imbalanced classification problems, a significant source of error in ML models. The imbalance occurs when there is a limited amount of the minority class in the dataset, which makes it difficult for a model to effectively learn the decision boundary [8]. In this research, we used the SMOTE technique to overcome this challenge by oversampling the examples in the minority class. We achieved this by producing duplicates of the minority class in the training dataset before fitting the model.
2. One-hot encoding technique helps convert categorical variables in a dataset into binary form so that the ML model will understand the data.
3. Normalization: The goal of normalizing data for ML models is to transform features and ensure that they are all on a similar scale. Normalization helps improve the training stability and performance of the model.
4. Exploratory Data Analysis (EDA) is a method of exploring the dataset to discover patterns, trends, and spot anomalies. Also, the dataset was cleaned at this stage to remove/handle missing or incomplete data by performing data imputation (substituting missing values with close estimations).
   On exploring the dataset, we found that;
   - The dataset contains more male applicants than female applicants.
   - The dataset contains more married applicants.
   - The dataset contains more applicants with good credit (1) than those with bad credit (0).

Furthermore, fig 1 below shows the correlation of key variables in the dataset and that Applicant_Income is the most positively correlated attribute to Loan_Amount.

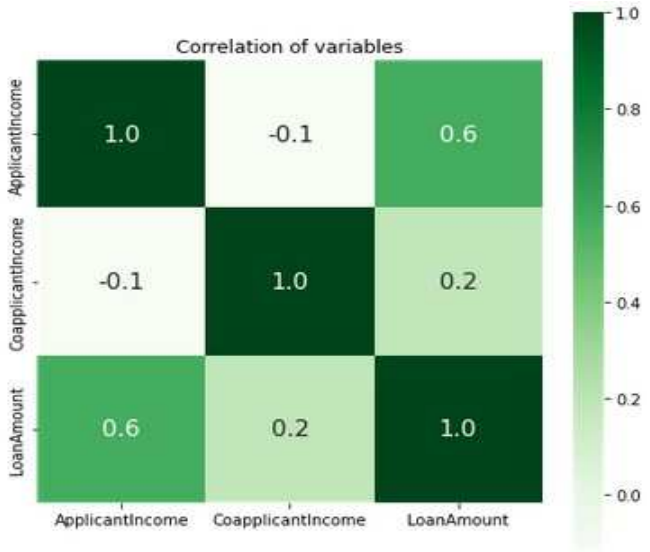


Fig 1: correlation of key variables in the dataset

IV. MODEL DEVELOPMENT AND RESULT

Evaluation metrics explain the performance of an ML model; the performance metrics used for this research include the Confusion Matrix and F1 Score.

The confusion matrix summarizes the number of correct and incorrect predictions by an ML model and breaks it down into classes:

i. "True positive" = Actual positive cases that the model correctly predicts.
ii. "False positive" = Actual positive cases that the model incorrectly predicts.
iii. "True negative" = Actual negative cases that the model correctly predicts.
iv. "False negative" = Actual negative cases that the model incorrectly predicts.

v. "Accuracy" = Overall correct predictions.

F1 score represents the mean of precision and recall values. The formula is given as follows:

$$2 * \frac{\text{precision} * \text{recall}}{\text{precision} + \text{recall}} \quad [9]$$

Where;

Precision = $\frac{\text{True positive}}{\text{True positive} + \text{False positive}}$ and Recall = $\frac{\text{True positive}}{\text{True positive} + \text{False negative}}$ [9]

A. Logistic Regression (LR) Algorithm

LR is a simple classification algorithm used to model a binary (0,1) variable. LR predicts the outcome of a response/dependent variable based on one or more other variables, called predictor/independent variable [10].

The logistic function is given as follows:

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad [11]$$

Where;

$1 + e$ denotes the exponential function.

β_0 is the intercept

$\beta_1 x$ is the regression coefficient

Fig 2 below shows the evaluation result of our LR model.

```
In [243]:
LRclassifier = LogisticRegression(solver='saga', max_iter=500, random_state=1)
LRclassifier.fit(X_train, y_train)

y_pred = LRclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
LRAcc = accuracy_score(y_pred,y_test)
print('LR accuracy: {:.2f}%'.format(LRAcc*100))
```

```
              precision    recall  f1-score   support

           0       0.79      0.75      0.77        20
           1       0.81      0.84      0.82        25

    accuracy                           0.80        45
   macro avg       0.80      0.79      0.80        45
weighted avg       0.80      0.80      0.80        45

[[15  5]
 [ 4 21]]
LR accuracy: 80.00%
```

Fig 2: LR model evaluation

B. K-Nearest Neighbor (KNN) Algorithm

KNN is a supervised ML algorithm that uses the Euclidean distance to calculate the distance between attributes and then matches the data points using the 'feature similarity' in the dataset. The formula is given as follows:

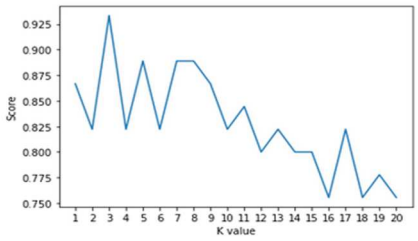
$$\text{dist}((x,y),(a,b)) = \sqrt{(x-a)^2 + (y-b)^2} \quad [12]$$

Where: (x, y) and (a, b) are the coordinates of two points in the plane.

Fig 3 below shows the evaluation result of our KNN model.

```
In [244]:
scoreListknn = []
for i in range(1,21):
    KNclassifier = KNeighborsClassifier(n_neighbors = i)
    KNclassifier.fit(X_train, y_train)
    scoreListknn.append(KNclassifier.score(X_test, y_test))

plt.plot(range(1,21), scoreListknn)
plt.xticks(np.arange(1,21,1))
plt.xlabel("K value")
plt.ylabel("Score")
plt.show()
KNAcc = max(scoreListknn)
print("KNN best accuracy: {:.2f}%".format(KNAcc*100))
```




```
KNN best accuracy: 93.33%
```

Fig 3: KNN model evaluation

C. Support Vector Machine (SVM)

The SVM algorithm explicitly seeks to find a hyperplane in an N-number of features that uniquely classify the data points (vectors) in a dataset [13]. Furthermore, the SVM algorithm is known for higher speed and better performance with a limited number of data samples.

It is given as follows:

$$w^* = \arg_w max \frac{1}{\|w\|_2} [min_n y_n \mid w^T(\phi(x) + b \mid] \quad [14]$$

Where: $[min_n y_n \mid w^T(\phi(x) + b \mid]$ represents the minimum distance of a point to the decision boundary, and $\arg_w max$ represents the maximum points of a function domain.

Fig 4 below shows the evaluation result of our SVM model.

```
In [249]:
SVCclassifier = SVC(kernel='rbf', max_iter=500)
SVCclassifier.fit(X_train, y_train)

y_pred = SVCclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
SVCAcc = accuracy_score(y_pred,y_test)
print('SVC accuracy: {:.2f}%'.format(SVCAcc*100))
```

```
              precision    recall  f1-score   support

           0       0.88      0.75      0.81        20
           1       0.82      0.92      0.87        25

    accuracy                           0.84        45
   macro avg       0.85      0.83      0.84        45
weighted avg       0.85      0.84      0.84        45

[[15  5]
 [ 2 23]]
SVC accuracy: 84.44%
```

Fig 4: SVM model evaluation

D. Decision Tree (DT) Algorithm

The decision tree algorithm uses the features/attributes present in a dataset to make informed decisions. The objective of the DT algorithm is to maximize the value of information gain [15]. It achieves this by splitting the features (nodes) starting with the highest information gain. It can be calculated using the below formula:

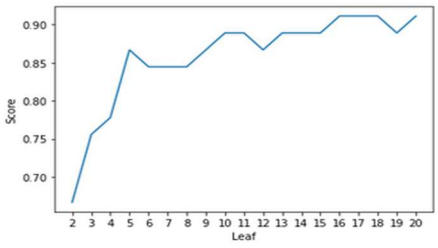
$$IG(T, a) = \mathrm{H}(T) - \mathrm{H}(T \mid a) \qquad [16]$$

Where: $\mathrm{H}(T \mid a)$ is the conditional entropy of T, and a is the value of the attribute.

Fig 5 below shows the evaluation result of our DT model.

```
In [248]:
scoreListDT = []
for i in range(2,21):
    DTclassifier = DecisionTreeClassifier(max_leaf_nodes=i)
    DTclassifier.fit(X_train, y_train)
    scoreListDT.append(DTclassifier.score(X_test, y_test))

plt.plot(range(2,21), scoreListDT)
plt.xticks(np.arange(2,21,1))
plt.xlabel("Leaf")
plt.ylabel("Score")
plt.show()
DTAcc = max(scoreListDT)
print("Decision Tree Accuracy: {:.2f}%".format(DTAcc*100))
```




```
Decision Tree Accuracy: 91.11%
```

Fig 5: DT model evaluation

E. Bagging and Boosting Algorithm

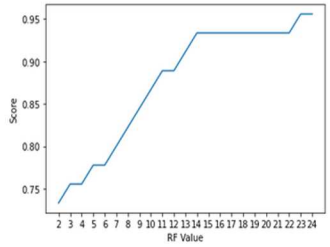
Bootstrap Aggregation (Bagging) is a very efficient ensemble technique that reduces ML models' variance. The Bagging technique achieves this by merging results from multiple classifiers modeled on the various sub-samples of a given dataset [17]. An example of the Bagging technique deployed for this research is the random forest (RF) Algorithm.

On the other hand, the boosting technique cuts across a special class of algorithms tasked with merging weak learners into strong learners. This is achieved by weighing the weak classifiers according to their accuracy and then iteratively learning and merging them into a final robust classifier [18]. The boosting techniques reweight the training set after every iteration and assign weights to any misclassified instances identified in the sequence [19]. In this research, the boosting technique deployed is the Gradient Boost (GBM) Algorithm.

Fig 6 below shows the evaluation result of our RF model.

```
In [250]:
scoreListRF = []
for i in range(2,25):
    RFclassifier = RandomForestClassifier(n_estimators = 1000, random_state = 1, max_leaf_no
des=i)
    RFclassifier.fit(X_train, y_train)
    scoreListRF.append(RFclassifier.score(X_test, y_test))

plt.plot(range(2,25), scoreListRF)
plt.xticks(np.arange(2,25,1))
plt.xlabel("RF Value")
plt.ylabel("Score")
plt.show()
RFAcc = max(scoreListRF)
print("Random Forest Accuracy:  {:.2f}%".format(RFAcc*100))
```




```
Random Forest Accuracy:  95.56%
```

Fig 6: RF model evaluation

Fig 7 below shows the evaluation result of our GBM model.

```
In [251]:
paramsGB={'n_estimators':[100,200,300,400,500],
      'max_depth':[1,2,3,4,5],
      'subsample':[0.5,1],
      'max_leaf_nodes':[2,5,10,20,30,40,50]}
GB = RandomizedSearchCV(GradientBoostingClassifier(), paramsGB, cv=20)
GB.fit(X_train, y_train)
GBclassifier = GradientBoostingClassifier(subsample=0.5, n_estimators=400, max_depth=4, max_
leaf_nodes=10)
GBclassifier.fit(X_train, y_train)

y_pred = GBclassifier.predict(X_test)

print(classification_report(y_test, y_pred))
print(confusion_matrix(y_test, y_pred))

from sklearn.metrics import accuracy_score
GBAcc = accuracy_score(y_pred,y_test)
print('Gradient Boosting accuracy: {:.2f}%'.format(GBAcc*100))
```

```
              precision    recall  f1-score   support

           0       0.90      0.95      0.93        20
           1       0.96      0.92      0.94        25

    accuracy                           0.93        45
   macro avg       0.93      0.94      0.93        45
weighted avg       0.93      0.93      0.93        45

[[19  1]
 [ 2 23]]
Gradient Boosting accuracy: 93.33%
```

Fig 6: GBM model evaluation

V. Discussion

Loan defaulting is a significant financial risk for the banking industry as it damages the interests of lenders and breaks the social trust. The academic field has put extensive effort into developing efficient machine learning techniques to help regulators carry out an accurate loan approval process in real-time. This research utilized state-of-the-art machine learning methods to build credible and accurate prediction models. Fig 7 below shows the comparison of all the machine learning algorithms deployed in this research. Our models achieved high-performance accuracy based on the precision and recall metrics, with the R.F. model achieving a 95% score.

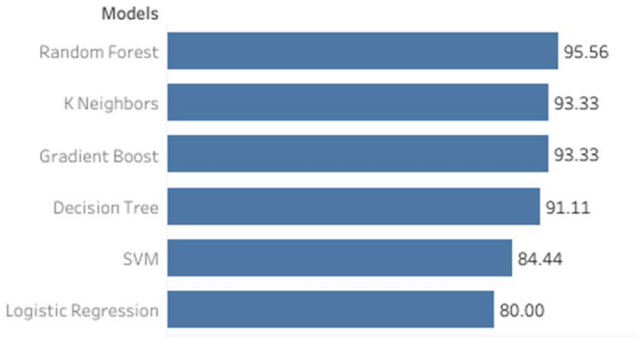


Fig 7: Model result comparison

VI. Conclusion

As more decision-makers in the financial industry seek to understand ways to improve their processes and maintain a balance between the security and reliability of their financial lending system, machine learning techniques can play a vital role in helping achieve this goal. Our ML models achieved high-performance accuracy in predicting loan eligibility in this research. We used the ensemble ML methods (bagging and boosting) and other techniques like SMOTE to ensure optimal predictive models. In general, the methods and algorithms deployed in this research could be instrumental to the successes of financial regulators, corporate, and individual borrowers in their effort to improve their overall loan approval process.

Additional Information

The datasets analyzed and complete documentation of the data analysis and model development process are available at: https://www.kaggle.com/orjiugochukwu/using-ml-algorithms-for-loan-approval-prediction.

References

[1] "Most commonly used A.I. application in investment banking worldwide 2020, by types." Statista, 15-Sept-2021 [Online]. Available: https://www.statista.com/statistics/1246874/ai-used-in-investment-banking-worldwide-2020/ [Accessed: 29-Jan-2022]

[2] G. Dorfleitner, E.M. Oswald, & R. Zhang. "From Credit Risk to Social Impact: On the Funding Determinants in Interest-Free Peer-to-Peer Lending." J Bus Ethics. 2021 Vol.170, pp. 375–400. https://doi.org/10.1007/s10551-019-04311-8

[3] A. Kumar, S. Sharma, & M. Mahdavi, "Machine Learning (ML) Technologies for Digital Credit Scoring in Rural Finance: A Literature Review." Risks 9.11 (2021): 192

[4] J. Xu, Z. Lu, and Y. Xie, "Loan default prediction of Chinese P2P market: a machine learning methodology." Scientific Reports, 2021, Vol. 11(1), pp. 1-19.

[5] H. Meshref, "Predicting Loan Approval of Bank Direct Marketing Data Using Ensemble Machine Learning Algorithms."International Journal of circuits, systems, and signal processing. 2020, Vol. 14, pp. 914-922 DOI: 10.46300/9106.2020.14.117

[6] A.S. Aphale, and S.R. Shinde, "Predict Loan Approval in Banking System Machine Learning Approach for Cooperative Banks Loan Approval." International Journal of Engineering Research & Technology (IJERT). 2020, Vol. 9 pp. 991-995

[7] "Loan Eligibility Dataset." Kaggle, 15-Aug-2020. [Online] Available:https://www.kaggle.com/datasets/vikasukani/loan-eligible-dataset

[8] A.S. Hussein, T. Li, C.W. Yohannese, & K. Bashir. "A-SMOTE: A new preprocessing approach for highly imbalanced datasets by improving SMOTE." International Journal of Computational Intelligence Systems. 2019, Vol. 12(2), PP.1412.of the 2003 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Stroudsburg, PA. 2003, pp. 129-136.

[9] D. M. Powers, "Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation." arXiv preprint arXiv:2010.16061 (2020).

[10] J.M. Hilbe. "Logistic Regression." International encyclopedia of statistical science. 2011, Vol 1: pp. 15-32.

[11] A. Saini. "Logistic Regression | What is Logistic Regression and Why do we need it?" 26-Aug-2021[Online] Available: https://www.analyticsvidhya.com/blog/2021/08/conceptual-understanding-of-logistic-regression-for-data-science-beginners/#h2_5 [Accessed: 28-Jan-2022]

[12] M. Shouman, T. Turner, and R. Stocker. "Applying k-nearest neighbour in diagnosing heart disease patients." International Journal of Information and Education Technology. 2012 Vol. 2(3), pp. 220-223.

[13] L.K. Ramasamy, S. Kadry, Y. Nam, & M.N. Meqdad. "Performance analysis of sentiments in Twitter dataset using SVM models. International Journal of Electrical and Computer Engineering (IJECE). 2021 Vol. 11, No. 3, pp.2275-2284 https://doi.org/10.11591/ijece.v11i3.

[14] R. Kunchhal. "Mathematics Behind SVM | Math Behind Support Vector Machine." 28-Dec-2020. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/10/the-mathematics-behind-svm/ [Accessed: 27-Jan-2022]

[15] K. Yadav, and R. Thareja. "Comparing the performance of naive bayes and decision tree classification using R." International Journal of Intelligent Systems and Applications. 2019, Vol.11(12), p.11.

[16] K. Ramya, Y. Teekaraman, & K.R. Kumar. "Fuzzy-based energy management system with decision tree algorithm for power security system." International Journal of Computational Intelligence Systems. 2019, Vol.12(2), pp.1173.

[17] L.G. Kabari, & U.C. Onwuka. "Comparison of bagging and voting ensemble machine learning algorithm as a classifier." International Journals of Advanced Research in Computer Science and Software Engineering. 2019, Vol. 9(3), pp.19-23.

[18] Z. Tian, J. Xiao, H. Feng, & Y. Wei. "Credit risk assessment based on gradient boosting decision tree." Procedia Computer Science. 2020, Vol.174, pp.150-160.

[19] "Bagging vs Boosting in Machine Learning." GeeksforGeeks. 07-Jul-2021[Online]. Available: https://www.geeksforgeeks.org/bagging-vs-boosting-in-machine-learning/ [Accessed 28-Jan-2022]