International Journal of Advances in Engineering and Management (IJAEM)
Volume 3, Issue 6 June 2021, pp: 985-992 www.ijaem.net ISSN: 2395-5252

Stroke Prediction Using Machine Learning

Vatsal S Chheda[1], Samit K Kapadia[2], Bhavya K Lakhani[3],Pankaj Sonawane[4*]

[1,2,3]*Student, Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India*

[4]*Assistant Professor,Department of Computer Engineering, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India*

Submitted: 25-05-2021 Revised: 01-06-2021 Accepted: 05-06-2021

ABSTRACT:A stroke is a brain attack, occurs when adequate blood flow to the brain is stopped, or a blood vessel in the brain ruptures. Several works have been carried out to accurately predict the outcome of various diseases by analyzing the various parameters associated with it and by making comparisons amongst the performances of different predictive data mining technologies. In our work, various classification algorithms are compared on the Stroke-Prediction-Dataset (Kaggle [1]), with features like hypertension, smoke status, BMI, etc. Here Scikit Learn's SimpleImputer has been used to fill in the missing values and ADASYN [6] is used to balance the imbalanced dataset. Random Forest [9,10,11] and XGBoost [13,16,18] are two classification models compared, with Random Forest's accuracy (97.67%) just edging out the XGBoost Classifier (96.894%).

KEYWORDS: Random Forest, XGBoost, Simple Imputer, ADASYN sampling, Stroke.

I. INTRODUCTION

Stroke, also called a cerebrovascular accident, CVA, or "Brain Attack", is the second leading cause of death globally. Strokes are of two types. The first being Ischemic stroke (part of the brain loses blood flow) and the second being Hemorrhagic stroke (bleeding occurs within the brain). Globally every 4th adult over the age of 25 is susceptible to have a stroke in their lifetime. In the United States, every year around 795,000 people have a stroke. About 610,000 people are the ones who are having a stroke for the first time. Around 87% of people have Ischemic Stroke in which the blood flow to the brain is blocked. Stroke is oneofthe leading causes ofgreatlong-term disability.Stroke is the cause of reduced mobility in more than half of the stroke survivors aged 65 and over. The impact of stroke is oftenshort- and long-term, counting onwhich part ofthe brain is affected and the wayquickly it'streated. Data generated from hospitals as patient records are an important source of information.

Using powerful data mining tools, it is possible to extract useful information from these records. The aim of the work is to take a specific area of medical research and help accurately predict the outcome based on specific attributes. The model will take the input as the patient's data and as output will provide the appropriate prediction. Extraction of hidden knowledge from the clinical database is done by the system, and it predicts whether the patient has the disease. The use of medical profiles, such as Hypertension, BMI, smoke status, etc. it is able to predict the chances of patients contracting a disease. Classification models, when provided with appropriate attributes can be used for accurate prediction of diseases. After going through various related and non-related research papers, it was shown that Random Forest Classifier always gave the best accuracy amongst all the other classification algorithms. So, the authors have used Random Forest Classification in their work and compared it with other classification algorithms with different feature selection methods. In our work, The Stroke Prediction Dataset (Kaggle [1]) is used for predicting whether a patient has the likeliness of having a stroke. This dataset includes patient's hypertension, BMI, and smoke status data.

Further, the paper is divided into five sections. Section 2 of the paper describes the review of the literature. Section 3 of the paper gives an understanding of our methodology, which consists of preprocessing, feature selection, sampling, and classification processes respectively. The analysis of the results is shown in Section 4 which is then followed by the dataset which we have used for our model is presented in Section 5 and which is followed by the conclusion in Section 6.

II. REVIEW OF LITERATURE

In [2], authors have proposed a model to predict the chances of stroke by using various machine learning algorithms such as Decision Tree, Naïve Bayes, and Neural Networks for predicting the likelihood of stroke occurrence. Here for dimensionality reduction Principal Component Analysis algorithm has been used. The dataset used here is the heart disease dataset provided by UCI. The dataset consists of Social Security Number, EKG (day/month/year), Age, Patient Number, Blood pressure, type of chest pain, Gender, Cigarettes, Family history, Hypertension, Nitrates, Cholesterols, Years, Heart rate, and calcium channel blocker. Firstly, pre-processing techniques are used to remove duplicate records, missing data, noisy and inconsistent data. Secondly, after preprocessing, the dimensions of the dataset are reduced using PCA which determines the attributes that are more involved in predicting the stroke disease. Lastly, three Classification algorithms were used for the diagnosis of patients with stroke disease.

In [3], the author's aim behind this work is to review the existing models for cardiovascular risk assessment. Their research provides in-depth analysis on various conventional methods and models like the Framingham model, Reynolds Risk Score, MUCA, PROCAM, and Global Vascular Risk Score. This research aims to review the models and examine the evidence available on new biomarkers and the non-clinical measures in improving the risk prediction.

In [7], the authors propose a novel automatic feature selection algorithm to predict stroke. The author uses the CHS dataset. The proposed algorithm in combination with SVM achieves a greater AUC score than the Cox potential hazard model. Further, the authors have designed a margin-based censored regression algorithm that combines the concept of margin-based classifiers with censored regression to achieve a better concordance index than the Cox model. The authors implement various algorithms to evaluate the prediction performance. This research demonstrates that their model can be used for identifying potential risk factors for diseases without per- forming clinical trials.

In [12], the authors have designed a model to predict a stroke's probability occurring using a neural network-based classification model and principal component analysis for dimensionality reduction. Three datasets are used here, Stroke & Claudication (STCL), Stroke & TIA(STTIA), and Stroke & Angioplasty (STAN). Feature selection is done using Decision Trees. This research has demonstrated that the ANN-based prediction of stroke occurrence yields an accuracy of 95%(STCL), 95.2%(STTIA), and 97.7%(STAN) respectively.

III. MODEL ARCHITECTURE

3.1 Data Visualization

Data Visualization is the process of describing the data in the form of graphical representation. Data Visualization can be used to find the trends and patterns in the dataset. The visual representation helps in analyzing the data quickly as compared to the raw data. It can also help in finding the flaws of the dataset. Data Visualization provides valuable insights into the data. To understand the data and select features for classification, this paper has used bar plots and a heatmap. The bar plot below represents the counts of the target variable (stroke.) This shows that the dataset is imbalanced and the classification will not be accurate. If the model is trained on this particular dataset, it will get overfit on class 0. The classification will be one-sided; hence, the model will not be useful in any circumstance.

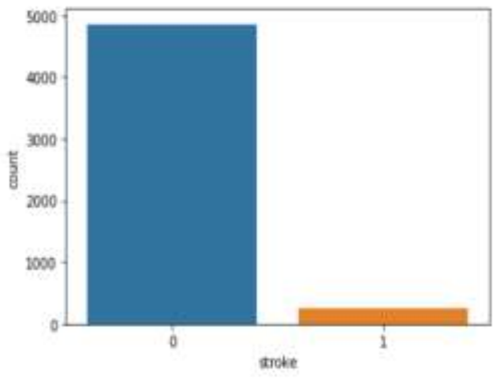

Fig 1: Bar plot for counts of two classes of stroke

Then this paper used a heatmap to understand the correlation between different features in the dataset. Heatmap is a statistical tool that is used to analyze the data. Heatmap is used to represent the correlativity amongst many features of the dataset. The correlation between the outcome and features can be very useful in determining which feature is very important. The values lie between -1 and 1. If the value is more towards one, it means that the feature is going to influence the output directly. If the value is towards -1, it means that the feature is going to influence the final result inversely, and if the value is closer to 0, it means that the feature is not likely to influence the final output much. The last column of the heatmap shows the correlation of a stroke with the other features of the dataset.

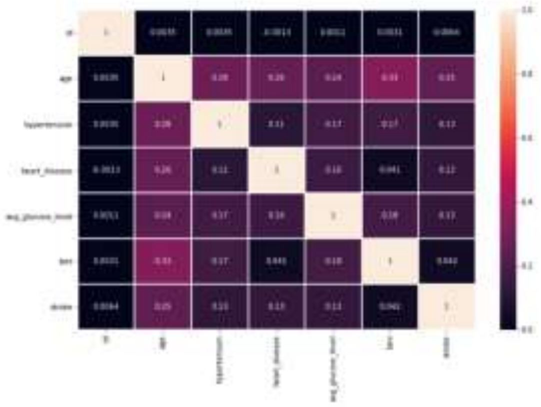

Fig 2: Heatmap

Now it is evident that the features are correlated to the Target Variable. Since all the features are correlated too a good amount, hence any feature cannot be discarded. So, after this Data Preprocessing can be applied to the Dataset.

3.2 Data Pre-processing

Upon Visualizing and performing Exploratory Data Analysis, this paper concluded that there were 201 records with the missing values of the Body mass Index were discovered. So, Scikit Learn'sSimpleImputer was used to replace these missing values with the mean Body Mass Index. The dataset had 5 Categorical Attributes: - gender, ever_married, work_type, Residence_type, smoking_status which needs to be encoded before being fed to the models. To resolve this issue OneHotEncoder[14] from the Scikit-Learn preprocessing module was used to encode them into numerical values. Standardization [17] is a scaling technique by which the values are centered around the mean with a unit standard deviation. That means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation. The resulting records now have 21 attributes. Here's the formula of standardization: -

$$X_{new} = \frac{X_i - X_{mean}}{\text{Standard Deviation}}$$

Fig 3: Standardization Formula

Two attributes: - Avg_glucose_level ranging from 55.12 to 271.74 and Body Mass Index ranging from 10.3 to 97.6. To resolve this, feature scaling [15,17] was applied to the dataset and was done before splitting the data into namely two categories, i.e., test set and train set, by using Standard Scaler from Sklearn's preprocessing module.

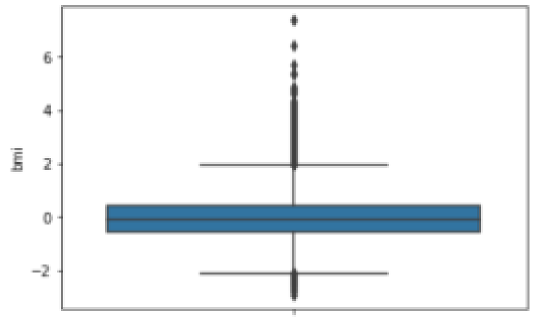

Fig 4: Box Plot of BMI before feature scaling

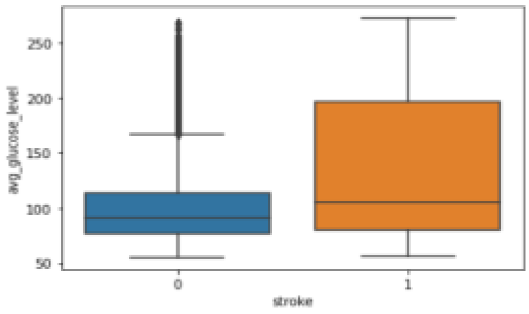

Fig 5: Box Plot of BMI after feature scaling

3.3 Sampling

Upon visualizing the data this paper had concluded that the dataset provided was imbalanced. It contained 4861 records with a stroke event value of 0 i.e. they never had a stroke, and 249 records with a stroke event value of 1 i.e. they have had a stroke in the past. In such cases, the concept of sampling is used to predict the output with greater accuracy. Sampling is one of the most efficient methods to overcome the problem of the imbalanced dataset. The goals of sampling are to create a dataset that has relative class distribution as in a dataset that has a somewhat equal number of distributions on either side, to create a clear decision boundary amongst the samples. Since sampling is used to increase the count of minority observations, the resulting dataset should be reasonable. It should be very much similar to the original dataset in many aspects i.e the distribution of the resulting dataset should be similar to that of the original dataset.

Sampling can be done in two ways-:
1)Oversampling- Oversampling is the process by which the number of minority class samples is increased in proportion to the majority class samples.
2)Undersampling- Undersampling is the process by which the number of the majority class samples are decreased in proportion to the minority class samples

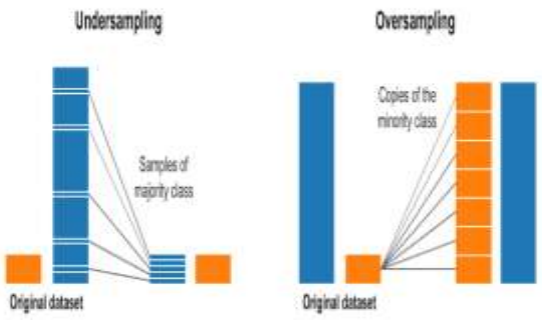

Fig 6: Types of Sampling

The concept of oversampling has been applied here. The reason behind it was that as the dataset had only 5000 samples, it made more sense to increase the number of the samples to attain better accuracy for the model.

Adaptive Synthetic Sampling Method for Imbalanced Data [19] (ADASYN): -

It is the improved version of SMOTE (Synthetic Minority Over-sampling Technique). This technique inherits the main weakness of SMOTE i.e. its ability to create inner point-outer point bridges. ADASYN also finds points that are outside of the homogeneous neighborhood. The algorithm uses Euclidean Distance for the KNN algorithm. The SMOTE algorithm is parameterized with K Neighbors.

The steps of the SMOTE algorithm are-:

1. A random minority point is selected
2. Then randomly K_neighbours nearest neighbors are selected which tend to belong to the same minority class.
3. The value of lambda(between 0 and 1) is randomly specified.
4. The new point is generated and placed on the vector between the two points which are located at the lambda percent from the original point

But the key difference in ADASYN is that it takes into account the distribution of density, which defines the number of synthetic instances produced for samples that are difficult to understand [20]. After sampling a considerable number of samples has been created for which the distribution of the target variable is equal. Now the dataset can be said to be balanced and this data can be provided to the classification models.

3.4 Models

Random Forest-: After sampling the dataset we have equal counts of each category and now the dataset is ready for classification algorithms. One such algorithm is Random Forest Classification [9].

The random Forest Classification algorithm belongs to the supervised learning field. It builds a forest by ensembling decision trees, and the model is usually trained with the bagging method. The intuition behind the bagging method is to combine learning models, which eventually increases the overall result. The random forest builds multiple numbers of decision trees on the data sample and then merges all of them by taking the average to provide a stable and accurate classification.

The random Forest Classification and Regression Algorithm works in the following two phases. In the first phase, it makes multiple decision trees and then combines all of them. In the second phase, it predicts each tree created in the first phase.[10]

Algorithm-:

1. Randomly select n data points from the training set.
2. Build the decision tree from the selected n points from the dataset.
3. To build the decision trees, provide the number of estimators.
4. Repeat step no.1 and step no.2.
5. For new data points, we'd like to seek out the predictions of every decision tree and can need to assign the new data points thereto a particular category that wins the bulk votes.

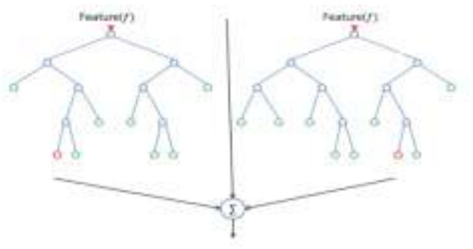

Fig 7: Random Forest

Random Forest instead of searching for the most important feature, searches for the most common amongst a subset of features. This is one of the reasons why this makes it a better model[9].

Hyperparameters [11] are used to increase the predictive power of the model and also to increase the speed of the model. **N_Estimators**is the most common hyperparameter which is used to increase the accuracy of the model. Basically, "n_estimators," tells the algorithm the number of trees it has to build. The greater the number of trees, the better the accuracy. As the number of trees is increased, there are chances that the computation becomes a bit slower.

Max_features is another important hyperparameter[11]. It provides the maximum number of features that random forest considers to split an internal node. This can help increase the accuracy of the model.

Random_state is also a hyperparameter that is used to increase a model's speed. Random_state is used to make the output of the model more predictable. When the value of random_state is the same, it will train the data on the same training data and hence the time required for computation decreases significantly thereby increasing the speed of the model.

Random Forest Algorithm is one of the most widely used algorithms for Classification and Regression Problems. The accuracy achieved by Random Forest is usually the best when compared to the other models. It also runs more efficiently on large databases when compared to the other algorithms[10].

XGBoostClassifier-: After using Random Forest Classifier on the dataset, the XGBoost classifier was also utilized to urge a far better accuracy[16]. XGBoost or eXtreme Gradient Boosting may be a sophisticated implementation of a gradient boosting algorithm. It's a supervised learning algorithm that attempts to impeccably predict a variable by combining the estimates of a gaggle of weaker and fewer complicated models.

Boosting is an ensemble method that seeks to make a robust classification model that supports weak classifiers. By adding a model on top of every other model iteratively, the errors of the older models are corrected by their succeeding predictor until the training data can be accurately predicted. XGBoost fits the new model to the new residuals of the previous prediction and then minimizes the loss that is done while adding the newest prediction, rather than assigning different weights to the classifiers after every iteration. So, in conclusion, the model is updated using gradient descent. Hence, it is called Gradient Boosting. XGBoost specifically, implements this algorithm for decision tree boosting with a further custom regularization term within the objective function.

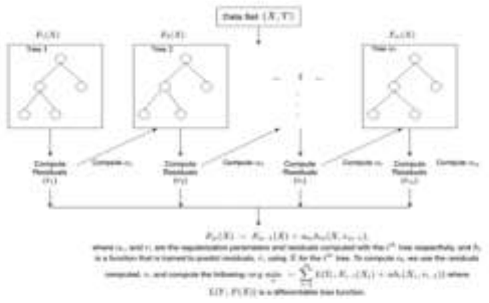

Fig 8: XGBoost Classifier

Hyperparameters-:

max_depth [default=6]

1. The maximum depth of a tree.'
2. If the value is increased, it may result in overfitting.
3. It can take values from 0 to infinity.
4. 0 is only accepted in loss guided growing policy when tree_method is set as hist or gpu_hist, and it indicates no limit on the depth
5. It can be tuned using cross-validation.

max_leaf_nodes

1. It is used to define the maximum number of terminal or leaf nodes in a tree.
2. This hyperparameter is often utilized in the place of max_depth. Since a binary tree is made, a tree with a depth of 'n' levels would produce a maximum number of 2^n nodes.
3. It has a default value of 0.

IV. RESULT AND ANALYSIS

Compare both the models: -

Model	**Accuracy**
XGBOOST	96.894%

Random Forest	97.67% (n_estimators = 200)

Random forest: -

n_estimators	Accuracy
200	97.67%

The above table illustrates the accuracy corresponding to the number of n_estimators. As the number of n_estimatorsincrease, the accuracy of the model also increases. The best accuracy was achieved when the value of n_estimators was 200. After a certain value of the n_estimators, the accuracy remained the same. Sometimes, using too many n_estimators can decrease the accuracy of the model.

XGB Confusion Matrix: -
Confusion Matrix-:

	0	1
0	952	20
1	40	920

Accuracy score -: 96.894%

V. DATASET

This research has used the stroke prediction dataset (available on Kaggle [1]). Dataset consists of 5110 records and 12 attributes per record. It contains Patient Id, Age, Gender, Hypertension, Heart Disease, Marriage Status, Work Type, Residence Type, Average Glucose Level, Body Mass Index, Smoking Status, and Stroke Event. Using this dataset, ten attributes are selected, which are associated with stroke risk factors used for prediction as described within the AHA guideline. This dataset has a Stroke Event value of 1 if the patient had a stroke and 0 if not.

VI. CONCLUSION

In this paper, two different approaches have been discussed to predict whether a person is likely to have a stroke or not. Random Forest Algorithm[9,10,11] and XGBoost Classifier[13,16,18] have been used to predict the results. The accuracy achieved from the Random Forest Algorithm came out to be **97.67%**. The hyperparameter provided was n_estimator. The value of the n_estimator varied from 5 to 100. The best accuracy was achieved when the value of n_estimator was 200. The accuracy achieved from XGBoostClassifier was **96.894%**. The model can predict by conducting only a minimal number of clinical trials. Features such as avg_glucose, smoking status, etc are to be given to the model for making a prediction. The outcome is likely to warn the patients hence, helping them to take precautions in advance to prevent them from having a stroke.

ACKNOWLEDGMENT

We would like to thank Prof. Pankaj Sonawane for his guidance throughout the paper. We would also like to thank Kaggle for providing the dataset for the paper.

REFERENCES

[1]. "Stroke Prediction Dataset", Kaggle.com, 2021. [Online]. Available: https://www.kaggle.com/fedesoriano/stroke-prediction-dataset. [Accessed: 02- Apr- 2021].

[2]. "(PDF) Prediction of Stroke Using Deep Learning Model", ResearchGate, 2021. [Online]. Available: https://www.researchgate.net/publication/320687273_Prediction_of_Stroke_Using_Deep_Learning_Model. [Accessed: 02- Apr- 2021].

[3]. Ijser.org, 2021. [Online]. Available: https://www.ijser.org/researchpaper/Stroke-

Prediction-Models-A-Systematic-Review.pdf . [Accessed: 02- Apr- 2021].

[4]. Arxiv.org, 2021. [Online]. Available: https://arxiv.org/pdf/1603.02754.pdf . [Accessed: 02- Apr- 2021].

[5]. Www3.nd.edu, 2021. [Online]. Available: https://www3.nd.edu/~dial/publications/hoens2013imbalanced.pdf . [Accessed: 02- Apr- 2021].

[6]. Ele.uri.edu, 2021. [Online]. Available: https://www.ele.uri.edu/faculty/he/PDFfiles/adasyn.pdf . [Accessed: 02- Apr- 2021].

[7]. People.csail.mit.edu, 2021. [Online]. Available: https://people.csail.mit.edu/khosla/papers/kdd2010.pdf . [Accessed: 02- Apr- 2021].

[8]. 2021. [Online]. Available: https://www.tutorialspoint.com/machine_learning_with_python/machine_learning_with_python_classification_algo . [Accessed: 02- Apr- 2021].

[9]. L. Forests, "Random Forests in Machine Learning | Random Forests for Data Science", Analytics Vidhya, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/12/lets-open-the-black-box-of-random-forests/ . [Accessed: 02- Apr- 2021].

[10]. "Machine Learning Random Forest Algorithm - Javatpoint", www.javatpoint.com, 2021. [Online]. Available: https://www.javatpoint.com/machine-learning-random-forest-algorithm. [Accessed: 02- Apr- 2021].

[11]. "Understanding Random Forest", Medium, 2021. [Online]. Available: https://towardsdatascience.com/understanding-random-forest-58381e0602d2 . [Accessed: 02- Apr- 2021].

[12]. "API Reference — scikit-learn 0.24.1 documentation", Scikit-learn.org, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/classes.html. [Accessed: 02- Apr- 2021].

[13]. J. Brownlee, "Extreme Gradient Boosting (XGBoost) Ensemble in Python", Machine Learning Mastery, 2021. [Online]. Available: https://machinelearningmastery.com/extreme-gradient-boosting-ensemble-in-python/ . [Accessed: 02- Apr- 2021].

[14]. J. Brownlee, "Why One-Hot Encode Data in Machine Learning?", Machine Learning Mastery, 2021. [Online]. Available: https://machinelearningmastery.com/why-one-hot-encode-data-in-machine-learning/ . [Accessed: 02- Apr- 2021].

[15]. "ML | Feature Scaling – Part 2 - GeeksforGeeks", GeeksforGeeks, 2021. [Online]. Available: https://www.geeksforgeeks.org/ml-feature-scaling-part-2/ . [Accessed: 02- Apr- 2021].

[16]. C. Python, "XGBoost Parameters | XGBoost Parameter Tuning", Analytics Vidhya, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/ . [Accessed: 02- Apr- 2021].

[17]. F. Standardization, "Feature Scaling | Standardization Vs Normalization", Analytics Vidhya, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/04/feature-scaling-machine-learning-normalization-standardization/ . [Accessed: 02- Apr- 2021].

[18]. "XGBoost Parameters — xgboost 1.4.0-SNAPSHOT documentation", Xgboost.readthedocs.io, 2021. [Online]. Available: https://xgboost.readthedocs.io/en/latest/parameter.html . [Accessed: 02- Apr- 2021].

[19]. J. Brownlee, "Tour of Data Sampling Methods for Imbalanced Classification", Machine Learning Mastery, 2021. [Online]. Available: https://machinelearningmastery.com/data-sampling-methods-for-imbalanced-classification/ [Accessed: 20- Apr- 2021].

[20]. Learning, "Imbalanced Classification | Handling Imbalanced Data using Python", Analytics Vidhya, 2021. [Online]. Available: https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/. [Accessed: 20- Apr- 2021].