Comparative Analysis of Machine Learning-Based Algorithms for Detection of Anomalies in IIoT

Bhupal Naik D. S., Vignan's Foundation for Science, Technology, and Research, India
Venkatesulu Dondeti, Vignan's Foundation for Science, Technology, and Research, India
Sivadi Balakrishna, Vignan's Foundation for Science, Technology, and Research, India

ABSTRACT

With the enormous increase in data, anomaly detection plays a prominent role in the finer analysis process. IIoT represents the industrial internet of things that at first chiefly alluded to a mechanical system whereby an enormous number of devices or machines are associated and synchronized using programming devices and third stage advancements in a machine-to-machine and internet of things, later an Industry 4.0. The data produced by multiple huge numbers of sensors are incredibly complicated, diverse, and massive in IIoT and is raw. These may contain anomalies which are needed to be identified for better data analysis. In this research, the authors compare the machine learning algorithms of classification for detecting anomalies. The algorithms being compared here are random forest (RF), logistic regression (LR), light gradient boosting machine (LightGBM), decision trees (DT), k nearest neighbors (KNN). Three IIoT benchmark datasets were taken into consideration for analysis. The results have shown that RF has outperformed other algorithms used for the detection of anomalies in IIoT data.

KEYWORDS

Anomalies, Classification, Industrial Internet of Things, Industry 4.0, Machine Learning, Outliers, Supervised Learning

I. INTRODUCTION

The word IoT can be defined as the correlation and coordination of diverse entities, where an entity could be an object, human, or machine which requests for or provides a service(J. Lin, Wei Yu, Nan Zhang, Xinyu Yang, 2017).In the world of industries, it is known as IIoT(Industrial Internet of Things). IIoT deals with the interconnection between machines, actuators, controllers and intensify productivity and automation in various industrial areas eg., transportation, manufacturing, and processing (A. Hassanzadeh, S. Modi& S. Mulchandani, 2015).Although IIoT proceeds to influence our current predicament and aim to create new future perspectives, it poses significant administrative and design problems (L. Da Xu, W. He&S. Li, 2014).

Anomaly is a term used to describe data that behaves in a way that is not intended, deviated from other data. The detection of anomalies (Anomaly Detection), also known as deviation detection, novelty discovery or outlier recognition, is identifying the patterns from the data that don't match the anticipated behaviour. Though anomaly is unusual, it is a significant phenomenon. Hence, the research

DOI: 10.4018/IJIRR.298647 *Corresponding Author

community has thus drawn considerable attention to anomaly detection (X. Liu & P. S. Nielsen, 2016) (M. Schreyer, T. Sattarov, D. Borth, A. Dengel, & B. Reimer, 2018). These anomalies can have a significant impact on the precision of classification based on data, prediction, and other operations; therefore, critical in identifying outliers rapidly in addition to effectively increase the accuracy of future operations based on data (Saihua Cai, Li Li, Sicong Li, Ruizhi Sun, Gang Yuan, 2020).

Anomaly detection has a broad range of applications; including the detection of fraud by credit card, industrial damage detection, healthcare, image processing, intrusion detection by computers, failure detection, and more (C. Chahla, H. Snoussi, L. Merghem& M. Esseghir, 2019).

Anomaly identification strategies in machine learning are classified into three categories depending on the labels available in the dataset:

Supervised Methods

ML models are designed for both abnormal and normal data in supervised learning techniques, in which unknown data case is given the label as anomalous or normal by assessing the concept it relates to (W. Cui & H. Wang, 2017).

Semi-Supervised Methods

Machine learning models are indeed to regular data in semi-supervised techniques, for which an unknown data example is labelled as ordinary if it is rational in following the model; else, the piece of instance is labelled as anomalous (W. Cui & H. Wang, 2017).

Unsupervised Methods

In unsupervised models, no training data is required, mainly because the anomalies in a given data set are assumed to be much more than normal data(W. Cui & H. Wang, 2017).

For this research, we developed a model for anomaly detection using multiple classification algorithms such as Logistic regression(LR), K Nearest Neighbor (KNN), Random Forest(RF), Decision Trees(DT), Light Gradient Boosting Machine (LightGBM), and compare the results based on various performance metrics.

This paper is assembled as follows: Section II discusses Literature Survey, Section III provides Methodologies, Section IV describes the Experimental evaluation, Section V discusses about the conclusion and future scope.

II. LITERATURE SURVEY

In this section, several aspects are examined concerning the relevant work, like IIoT, several anomaly detection machine algorithms.

Emiliano Sisinniet al. (Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, Mikael Gidlund, 2018) have presented a detailed study on IoT, IIoT, and Industry 4.0 including the opportunities and the challenges in this paradigm. They also have discussed some of the recent works in research to overcome the challenges that involve the need for energy efficiency, interoperability, security, real-time performance, and privacy in the IIoT field.

Mahmudul Hasan et al.(Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem, 2019) provided a comparison of different algorithms namely Artificial Neural Networks(ANN), Logistic Regression(LR), Decision Trees(DT), Support Vector Machine(SVM), Random Forest(RF) to discover attacks and anomalies in the IoT systems. They concluded that RF has shown better performance with an accuracy of 99.4% compared to others and it is a good technique to use on such problems where cyber attacks need to be detected.

Rui Zhu et al. (Rui Zhu, Xiaoling Ji, Danyang Yu, Zhiyuan Tan, Liang Zhao, Jiajia Li, Xiufeng Xia, 2020)suggested a new GAAOD architecture to enable KNN-based outlier detection of IoT

streaming data (grid-based approximate average outlier detection). The grid file-based index first was suggested for the management of the window streaming data. The next suggested an approximately KNN search to respond through a novel algorithm. Thirdly, they suggested a method focused on the k-sky band to maintain a window for candidates. The accuracy and precision of the proposed algorithm were checked by theoretical study and observation.

Di Wu et al.(Di Wu, Zhongkai Jiang, Xiaofeng Xie, Xuetao Wei, Weiren Yu, Renfa Li, 2019) suggested the LSTM-Gauss-NBayes system, a long short-term neural memory network (LSTM-NN) synergy, with the outlier IIoT model for Gaussian Bayes. They concluded that their experiments showed promising results.

Imran Razzak et al.(Imran Razzak, Khurram Zafar, Muhammad Imran, Guandong Xu, 2020) suggested a new anomaly discovery model for large-scale data in their study. For this model, random nonlinear functionality is used in support of bounded loss function vector machines instead of seeking optimized support vectors of the unlimited loss function. They concluded that their findings were more accurate than prior studies on ten benchmarkdatasets.

Nashreen Nesa et al.(Nashreen Nesa, Tania Ghosh, Indrajit Banerjee, 2018) proposed a sequence-based learning approach for outlier detection that works for both Error and Event. The algorithm is modelled as a parametric non-distributive algorithm and operates well for limited training sets, which is a requirement for all IoT objects. Simulations are made using few benchmarking datasets, a medical data set, and an experimental testbed in the real world. The results show very high accuracy, with error detection up to 99.65% and event detection 98.53%.

Xiaodan Xu et al. (Xiaodan Xu, Huawen Liu, Minghai Yao, Li Li, 2018) addressed some common outlier detection issues for high-dimensional data and tried to provide an overview of cutting-edge outlier detection strategies for high-dimensional data. In addition, they conducted a comprehensive public data sets experiment to test the popular outlier detection approaches. During the tests, the collection of data and various measurement measures were addressed for outsourcing identification. They also compared the efficiency of various methods on a broad range of data sets by taking into account the most widely used calculation and investigated the use of these standard outlier methods.

Osama Abdelrahman and Pantea Keikhosrokiani (Osama Abdel Rahman and Pantea Keikhosrokiani, 2020) examined assembly data for two series of products to identify and diagnose the potential reasons for anomalous data points. They used various strategies for the identification of anomalies, including HBOS, IFOS, KNN, CBLOF, LOF, OCSVM, and ABOD. For 54132data points with ABOD, there were 62 anomaly data points and with the 54104 data points with KNN algorithms, there were 343 anomaly data points, with no clear over eliminate existence in both sequences assemblage machines.

Haomiao Yang et al. (Yang, H., Liang, S., Ni, J., Li, H., & Shen, X, 2020) suggested for intelligent industrial control systems, a distributed kNN classification algorithm (SEED-KNN) which is reliable and coherent. They have built-in particular a new VHE system that meets semantic and syntactic security and high performance in public storage and vector encryption. Using VHE it is proposed that SEED-kNN efficiently classifies encoded massive data on multiple devices based on similarity values. They concluded that the proposed algorithm can be used in a range of applications, for example, faulty component recognition and classification, and anomaly detection, for sparse industrial control systems.

To detect anomalies in aging IIoT, Bela Genge et al. (BelaGenge, Piroska Haller, Calin Enachescu, 2019) developed a model based on multivariate statistical analysis PCA, alongside Hotelling's T 2 statistics, and the univariate cumulative sum. The identification of stealth attacks seeking to detect the data set at each age is a revolutionary aspect of the development strategy. They concluded that the algorithm applies to IIoT adds superior efficiency to the new techniques as seen by the comprehensive experimental findings on a Continuous Stirred Retank Reactor (CSTR). Table 1 depicts the summary of various ML techniques in the discovery of anomalies in IIoT data.

The following list of limitations are observed or inferred from the literature survey

Table 1. Summary of the various ML techniques over anomaly detection

Author Names/Year	Technique/s used	Data	Conclusions drawn
Emiliano Sisinni, Abusayeed Saifullah, Song Han, Ulf Jennehag, Mikael Gidlund, 2018	ML, DL, NN	Industrial data, IIOT	Survey to overcome challenges, increase energy efficiency.
Mahmudul Hasan, Md. Milon Islam, Md Ishrak Islam Zarif, M.M.A. Hashem, 2019	ANN, LR,DT,SVM,RF	IoT network data to detect errors and attacks	RF has shown better performance with an accuracy of 99.4%
Rui Zhu, Xiaolingh Ji, Danyang Yu, Zhiyuan Tan, Liang Zhao, Jiajia Li, Xiufeng Xia, 2020	GAAOD architecture to enable KNN-based outlier detection of IoT streaming data	IoT streaming data	Achieved 94% accuracy and 92% precision results compared with the existing works
Di Wu, Zhongkai Jiang, Xiaofeng Xie, Xuetao Wei, Weiren Yu, Renfa Li, 2019	LSTM-Gauss-NBayes system	IIoT data	Concluded that their experiments showed promising results
Imran Razzak, Khurram Zafar, Muhammad Imran, Guandong Xu, 2020	Random nonlinear functionality is used in support of bounded loss function vector machines	Large scale data using ten bench marked datasets.	Concluded that their findings were more accurate than prior studies
Nashreen Nesa, Tania Ghosh, Indrajit Banerjee, 2018	sequence-based learning approach for outlier detection	IoT medical data set, experimental testbed in the real world	The results show very high accuracy, with error detection up to 99.65% and event detection 98.53%
Xiaodan Xu, Huawen Liu, Minghai Yao, Li Li, 2018	cutting-edge outlier detection strategies for high-dimensional data	High-dimensional public datasets	Compared the efficiency of various methods on a broad range of data sets
Osama Abdel Rahman and Pantea Keikhosrokiani, 2020	HBOS, IFOS, KNN, CBLOF, LOF, OCSVM and ABOD	assembly data for two series of products	ABOD and KNN gave good results
Yang, H., Liang, S., Ni, J., Li, H., & Shen, X, 2020	a distributed kNN classification algorithm (SEED-KNN)	IIoT data	Concluded that the proposed algorithm can be used in a range of applications
BelaGenge, Piroska Haller, Calin Enachescu, 2019	Multivariate statistical analysis PCA, Hotelling's T^2 statistics, Univariate cumulative sum	Aging IIoT on Continuous Stirred Retank Reactor (CSTR)	Concluded that the algorithm applies to IIoT adds superior efficiency to the new techniques

- As it is observed from the survey, there are very few researches that are based on IIoT datasets.
- The data produced in IIoT is very huge which leads to maintenance difficulty.

This gave us the motivation leading to the interest in taking up this research, where IIoT is the latest technology that is being utilized and satisfactory analysis of data needs to be performed.

III. METHODOLOGIES

In this section, we discuss the evaluation process of our model and the algorithms used for training the model. For validating the model k-Fold Cross-validation technique is applied. The detailed description of k-cross validation is given below after the study of the algorithms.

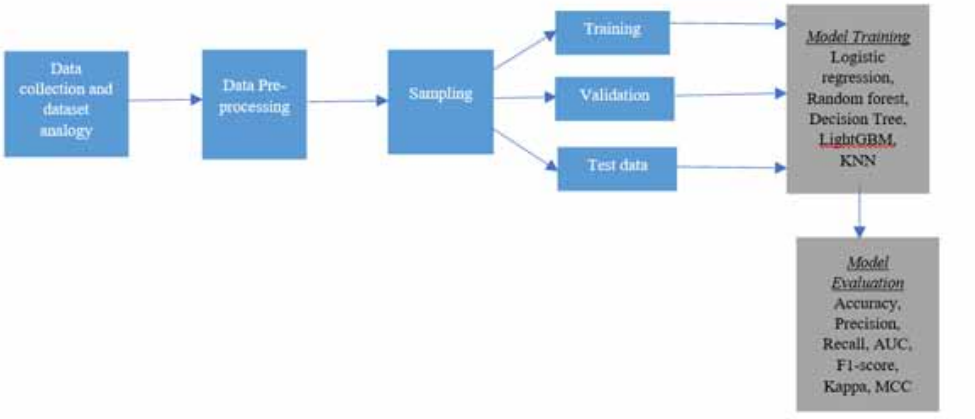
Fig 1 shows the framework of our model, the evaluation process. As it is shown the detailed overview is given below:

i) *Data Collection*: The first step in the model includes the collection of datasets.
ii) *Data Pre-processing*: The second step does the pre-processing which includes handing the missing values, dimensionality reduction, and vectorization, which is used to speed up the computation.
iii) *Sampling*: Data sampling refers to statistical techniques used for the estimation of population parameters by selecting samples from the dataset. The techniques used for sampling are Random sampling, Stratified sampling.
iv) *Splitting the data*: In this phase, the data is divide into a training set and testing set. By default, ratio of training to testing split is 70:30(percent) respectively., i.e., 0.7 data is taken into training and 0.3 is taken into testing.

training set—a subset to train a model.
test set—a subset to test the trained model.

v) *Learning algorithms*: The algorithms that were used for training our model are discussed in detail below.
vi) *Evaluation of the model*: Once the training is done the model is evaluated and the testing set is evaluated later based on how the model is trained. The performance can be measured using performance metrics that were discussed in the Experiment Evaluation section.

Figure 1. *Framework for detecting anomalies in IIoT*



The algorithms that were taken into consideration will be discussed below.

For validating the model k-cross validation technique is used. A detailed description of k-cross validation is given below.

A) Logistic Regression

Logistic regression is a classification method based on a supervised technique that is used to estimate the possibility of a target attribute. The nature of the dependent variable or target is binary or arbitrary, meaning only two classes are feasible. In other words, the classes are either 0 (NO/failure) or 1 (YES/success).

The LR is applicable in a wide range such as Cancer detection, Spam detection, Fraud detection, Anomaly detection, etc. The difference between Logistic and Linear regression is that Logistic regression is used in classification, unlike linear regression which is used for regression problems.

Instead of applying a regression line, we employ a "S" form logistic function that estimates two peak values in logistic regression (0 or 1).

Mathematically it can be denoted as:

$$y = \frac{1}{1 + e^{-x}}$$

were
y=dependent variable or target,
e=euler's number,
x=input variable or independent variable.

B) Decision Trees

Decision Tree is the technique of supervised learning, which is used for both Regression and classification; however, it is often used to solve classification problems. The classification is tree-structured, where inner denotes the characteristics of a set of data, branches indicate decision rules and each leaf node denotes the results. It is a schematic representation to get all the potential results to a decision/problem on given criteria.

The DT consists of two nodes namely the Leaf node and Decision node. Decision nodes are used for making a decision or test, which are obtained based on the data features, may have several branches, while the leaf nodes are the results of those decisions and do not contain any further branches.

This algorithm was given the name decision tree. As like a tree, it begins with the Root Node expanding over additional branches and building a structure like a tree. Classification and Regression Tree algorithm (CART) is used to build a tree. The decision tree poses a query, and it splits the tree further into sub-trees depending on the answer (Yes/No).

C) Random Forest

Random Forest is a well-known supervised learning algorithm that is based on ensemble learning notion, is a method in which multifarious classifiers are amalgamated to address a complex issue and to enhance model efficiency.

Random Forest is a classifying algorithm, as the name implies, containing multiple decision trees on different sections of the dataset and taking the average step of increasing the prediction accuracy of the dataset. The Random Forest collects the estimate from each tree and is dependent on the majority of votes on its assumptions, instead of depending on one Decision Tree, and forecasts its end result. The larger number of forest trees contribute to greater accuracy and avoid the congestion problem.

D) Light Gradient Boosting Machine

LightGBM is a decision tree-based gradient boost model to achieve the better performance as well as reduce memory use. Two new techniques are used namely Exclusive Feature Bundling (EFB) and Gradient-based One Side Sampling which overcomes the disadvantages of algorithm based on

histograms used mostly in all GBDT implementations. The EFB and GOSS are designed to make the model effective and to give it a cutting edge concerning additional GBDT modules.

Unlike many other algorithms based on Boosting where the tree grows level-wise, the tree in LightGBM is divided leaf-wise. The leaf having utmost delta loss is chosen to grow. As the leaf is fixed, the algorithm based on leaf-wise split has minimal loss contrary to the level-wise split algorithm. However, in small datasets, the tree grown Leaf-wise might maximize the model's difficulty and might lead to overfitting.

E)K-Nearest Neighbors

K-Nearest Neighbor is a classical Machine Learning algorithm that uses the Supervised Learning approach. K-NN assumes the new data/case and available data's similarity and the new data is put into the data into the most comparable group.

In K-NN, all the data available is stored and categorizes a new similarity data point. i.e; a new data can quickly be categorized into a well-suited group using the K- NN algorithm as fresh data emerges. For classification and regression, K-NN algorithm may be used, although it is mainly used for classification issues. It is a non-parametric-based algorithm, meaning that the underlying data is not inferred.

K-NN is a lazy learner algorithm as it executes an action at the time of classification on the dataset without immediately learning from the training data. The dataset is just stored in the phase of training and the newly arrived data is classified into a group depending on the similarity measures with the available data.

F) K-Fold Cross Validation

To assess the competencies of machine learning models, a statistical method namely the cross-validation approach can be used. Applied machine learning involves comparing and adopts a strategy for a predictive modelling challenge because it is simple to decipher, easy to deploy, and leads in capabilities that are usually less inclined than other approaches.

Cross-validation is a resampling process performed on a small selection of data to assess machine learning methods. A single parameter named k is used to indicate the variety of groups to be divided by a given sample data. So, k-fold cross-validation is typically termed. A value for k is estimated, it could be used instead of k in the model reference like k=10 which is 10-fold cross-validation.

The basic approach of K-Fold cross validation is as follows:

1. Re-arrange the dataset arbitrarily.
2. Cleave the dataset into k categories.
3. For every distinct group:
    a. Choose the block as a hold or test set of data.
    b. Consider the other categories as a data set for training.
    c. Build a method for the training data and validate it on the test data set.
    d. Hold the assessment score and exclude the model.
4. Resume the model's ability with the model assessment sample scores.

Salient point is that every insight in the sample data is allocated to a certain category and remains for the course of the procedure in that group. This allows every instance to be used in the specified time and used for the k-1-times model training. The outcomes of a k-fold cross-validation process are generally summed up as the average of the model competencies.

IV. EXPERIMENTAL EVALUATION

A) Datasets

Three datasets were taken into consideration which are available in KAGGLE. Each dataset is described below.

1) E-Coating Ultrafiltration Maintenance Dataset

This dataset contains data recorded in 15 days from an IIOT system in an electrophoresis painting plant. The dataset contains 9 attributes containing a total of 720 records which can be seen in table 2

Table 2. Dataset Description

ATTRIBUTES	DESCRIPTION
TIME	Timestamp
FM1	Flow meter 1 - Ultrafiltration subsystem
PE1	Pressure 1 - Input pressure for ultrafiltration subsystem
PE2	Pressure 2 - Output pressure for ultrafiltration subsystem
PE3	Pressure 3 - Input pressure for circulation subsystem
PE4	Pressure 4 - Output pressure for circulation subsystem
TP1	Temperature 1 - at the paint tank
TP2	Temperature 2 - at the radiator of circulation subsystem
EPOCH	Epoch of timestamp

2) Semiconductor Manufacturing Process Dataset

The dataset is obtained from the semiconductor manufacturing process. there are 1567 instances and 591 attributes. The manufacturing process unit is under reliable surveillance via monitoring of signals/variables obtained from various sensors. Labels are presently represented as Pass/Fail. 591 attributes include Time and class label Pass/Fail columns.

Attributes were not described and they were assigned numbers(0 to 9) as names.

3) Demand Vs Response Data For Iot Analytics

Industrial demand /response IoT data for IoT analytics. This dataset contains 7 attributes with 16382 records.

Attributes

DEMAND_RESPONSE, Area, Season, Energy, Cost, pair no, Distance

4) High Storage System Data for Energy Optimization

The high storage system includes four short conveyor belts and is used to transmit a standalone executable among two positions. This dataset is taken from a Smart factory in Lemgo. Failure of a belt results in the packet transportation failure too. It consists of 20 attributes with 23645 records.

B) Experiments

All the implementation is done using mainly PyCaret Library, which is a low-code, open-source machine learning library in Python. The processing was done in Jupyter notebook on an i5 processor with 8GB memory with Windows 10.

C) Performance Metrics

1) *Accuracy*:

This has been the major efficient metric used in classification algorithms. This shows the ratio between number of true predictions and all the made predictions. It can be denoted as

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

2) Precision:

Precision is defined as number of TPs predicted that belongs to the actual positive class and is given as

$$\text{Precision} = \frac{TP}{TP + FP}$$

3) Recall:

This is the number of predicted positive classes made out of the all the predicted results and is calculated as below:

$$\text{Recall} = \frac{TP}{TP + FN}$$

4) F1 Score:

This provides the harmonic mean of both recall and precision and is obtained by

$$\text{F1} = 2 * \left(\frac{\left(precision * recall\right)}{\left(precision + recall\right)} \right)$$

F1 score is 1 means the best and 0 being the worst values.

5)AUC (AREA UNDER ROC CURVE):

The ROC-Receiver operating characteristic curve is the probability curve and AUC is the separability curve. In other words, the AUC-ROC metric will inform the ability of the model to distinguish between classes. When the AUC is larger, the model performs better.

6) KAPPA:

Cohen's Kappa is a metric to measure the efficiency of two raters who score the same amount and to determine how often the raters approve.

7) MCC:

Matthews's correlation coefficient (MCC) or phi-cofactor is used to calculate the consistency of binary classifications in machine learning.

$$\text{MCC} = \frac{\left(TP \times \ TN\right) - \left(FP \times FN\right)}{\sqrt{\left(TP + FP\right)\left(TP + FN\right)\left(TP + FP\right)\left(TP + FN\right)}}$$

D) Results And Analysis

The graphs that are used in the evaluation of the algorithms are discussed below:

1) *ROC curve*: The Receiver Operating Characteristic curve, or ROC curve, is a powerful method for estimating the possibility of a binary outcome. It depicts a graph comparing the x-axis indicating false positive rate and the y-axis indicating negative rate for a variety of candidate threshold values ranging from 0.0 to 1.0.A no-skill classification model cannot distinguish among classes and will only determine a random or persistent class. In such a case the plot contains a diagonal line from the lower left corner to the upper right corner at each threshold and has an AUC of 0.5.

A classifier with perfect skill is depicted with a line going from the lower-left corner to the upper left corner and then across to the upper right corner.

2) *Confusion matrix:* It is a matrix of NxN size where the comparison between values of the actual target class and predicted target class values is depicted, where N is several class labels. It gives us an integrated perspective of how well the classifier executes and what types of errors it produces.

TP- True Positive – The actual positive values predicted as positive.
FP- False Positive – The actual negative values predicted as positive.
FN– False Negative– The actual positive values are predicted as negative.
TN- True Negative– The actual negative values predicted as negative.
This indicates that FP and FN represent the values falsely predicted.

3) *t-SNE Manifold Plot*: It is similar to PCA, which is a dimensionality reduction technique that simplifies datasets graphically. It is capable of clustering data stores based on current determinations over a huge proportion of samples.
4) *Calibration curves:* Calibration curves are used to assess how well a classification algorithm is trained, how the possibilities of interpreting each class label differentiate. The expected average

outcome from each bin is represented by the x-axis. The proportion of positives is shown on the y-axis

The results of all the five algorithms for each dataset are provided below:

1) E-coating Ultrafiltration Maintenance Dataset

The comparison of five algorithms and the result plots were given below for this dataset. Table 3 gives the comparison of the five classifiers used with all the performance metrics. Here, RF, DT, Light GBM, KNN got similar results with less difference in terms of accuracy. But other metrics should also be considered. Accordingly, RF has shown best results.

Table 3. Comparison of five classifiers over E-coating ultrafiltration dataset

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.9861	0.9703	0.7000	0.9000	0.7633	0.7590	0.7774	0.1650
dt	Decision Tree Classifier	0.9841	0.8490	0.7000	0.8667	0.7433	0.7380	0.7581	0.0140
lightgbm	Light Gradient Boosting Machine	0.9801	0.9812	0.7333	0.8067	0.7305	0.7231	0.7436	0.0310
knn	K Neighbors Classifier	0.9801	0.9226	0.7333	0.7667	0.7233	0.7155	0.7295	0.0240
lr	Logistic Regression	0.9544	0.6597	0.0000	0.0000	0.0000	0.0000	0.0000	0.0200

Figure 2(a). Applying Random Forest =(a)ROC curves

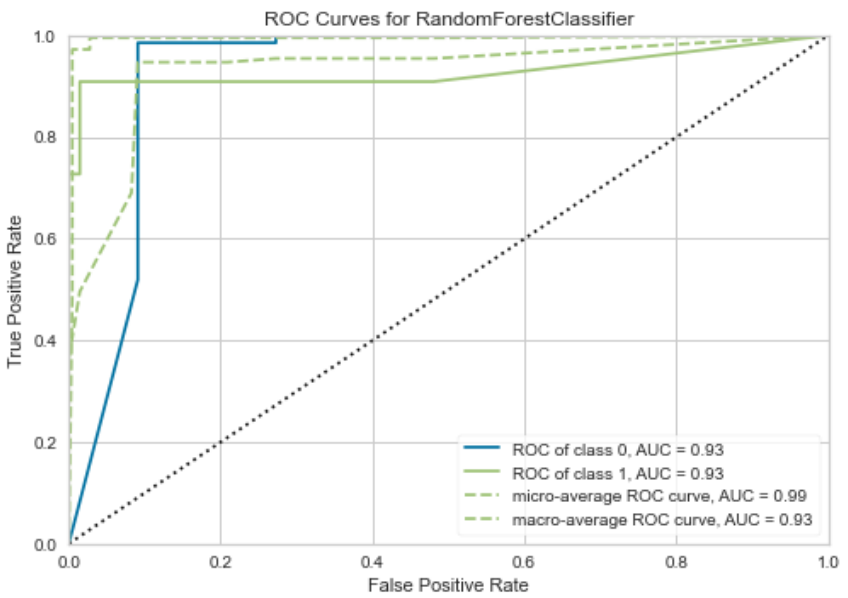


Figure 2(b). Applying Random Forest = (b) Confusion Matrix

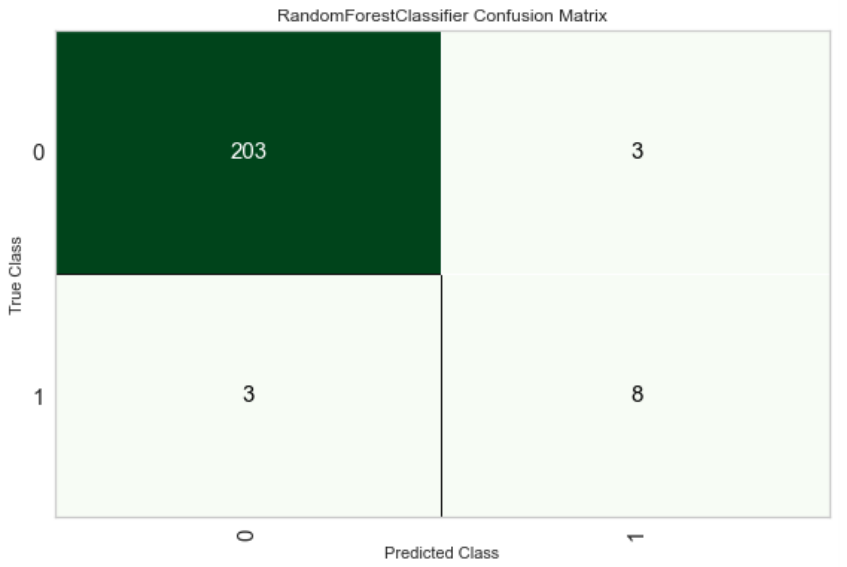


Figure 2(c). Applying Random Forest = (c) t-sne manifold curves

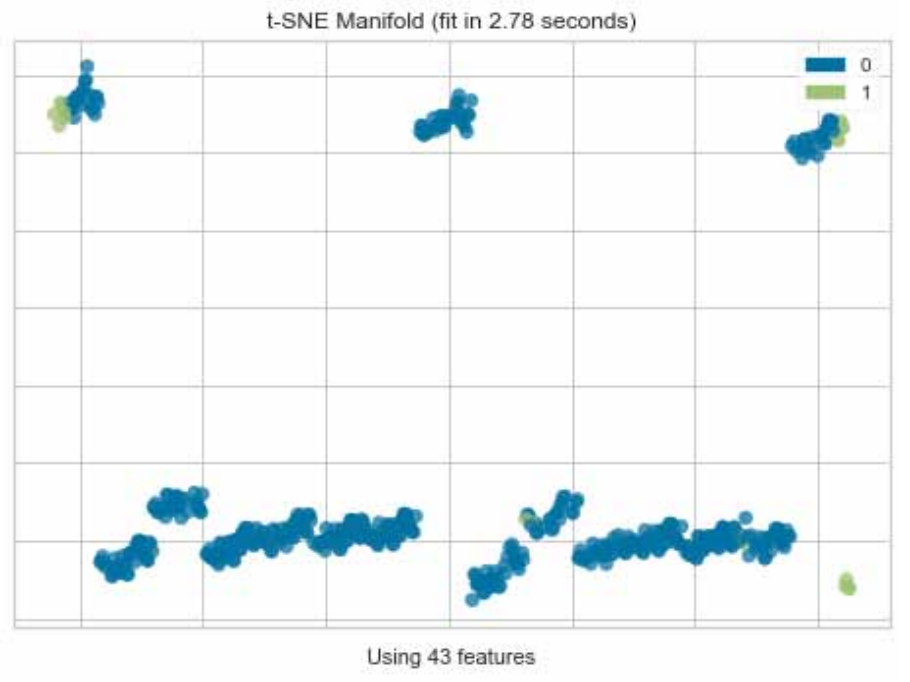


Figure 2(d). Applying Random Forest = (d) Calibration curve

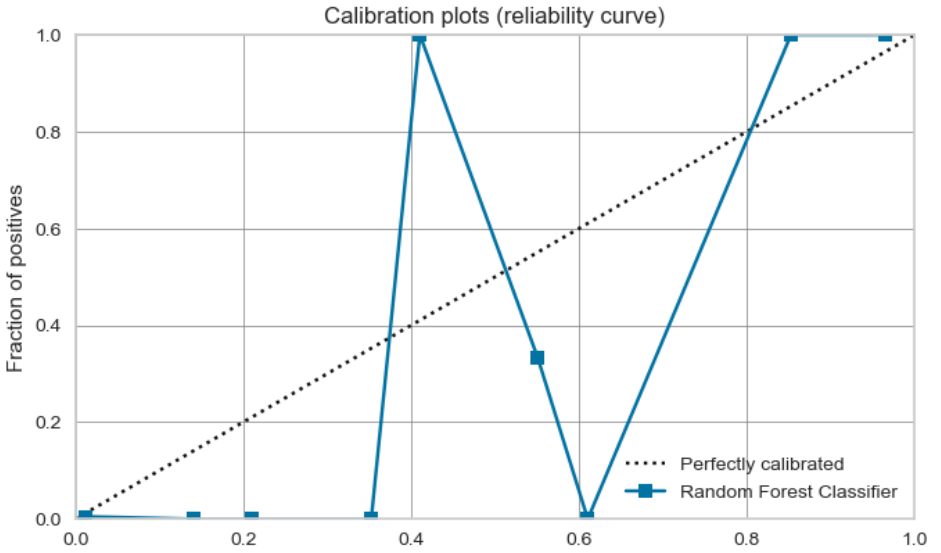


Figure 3(a). Applying Decision Tree (a)ROC curves

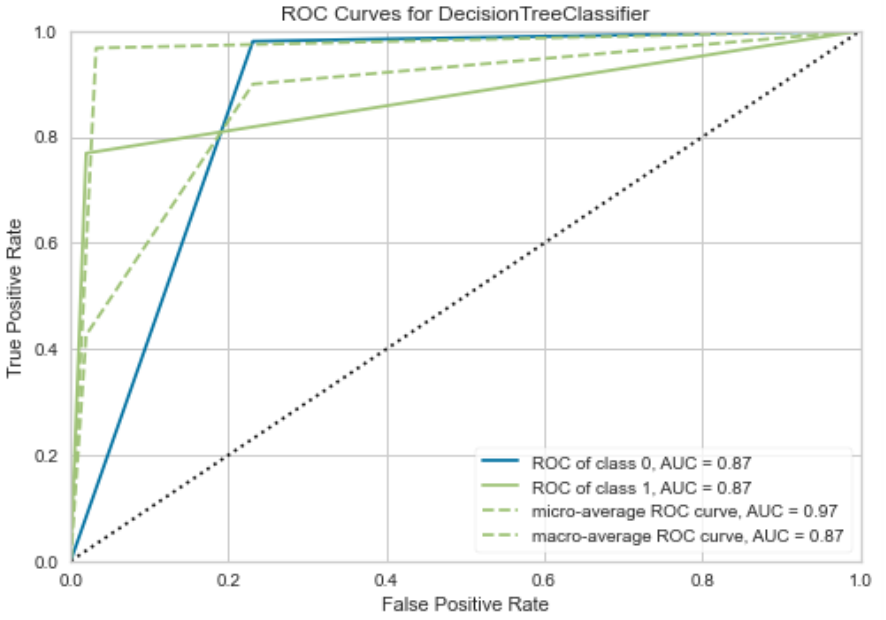


Figure 3(b). Applying Decision Tree (b) Confusion matrix

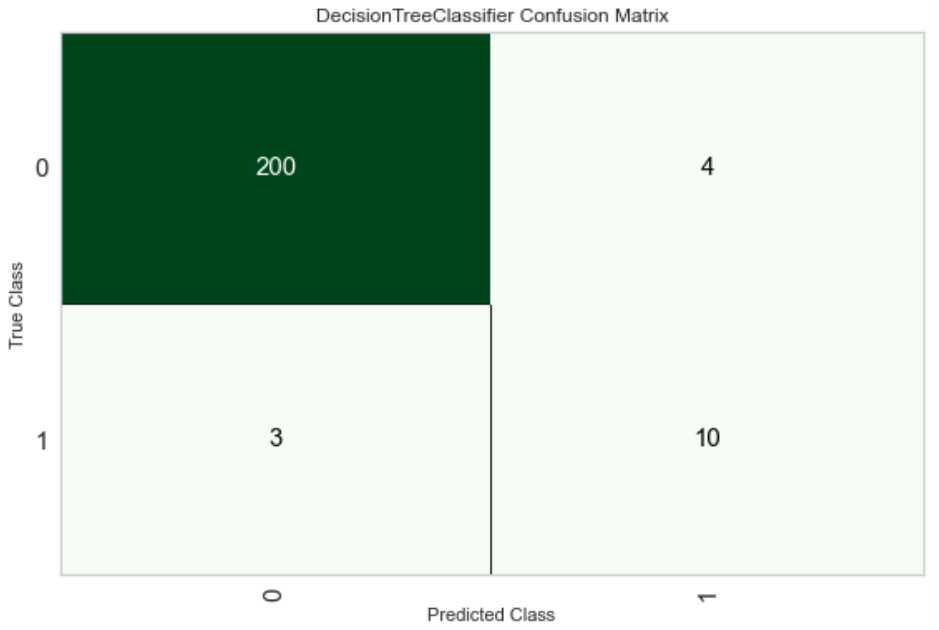


Figure 3(c). Applying Decision Tree (c) t-sne manifold curves

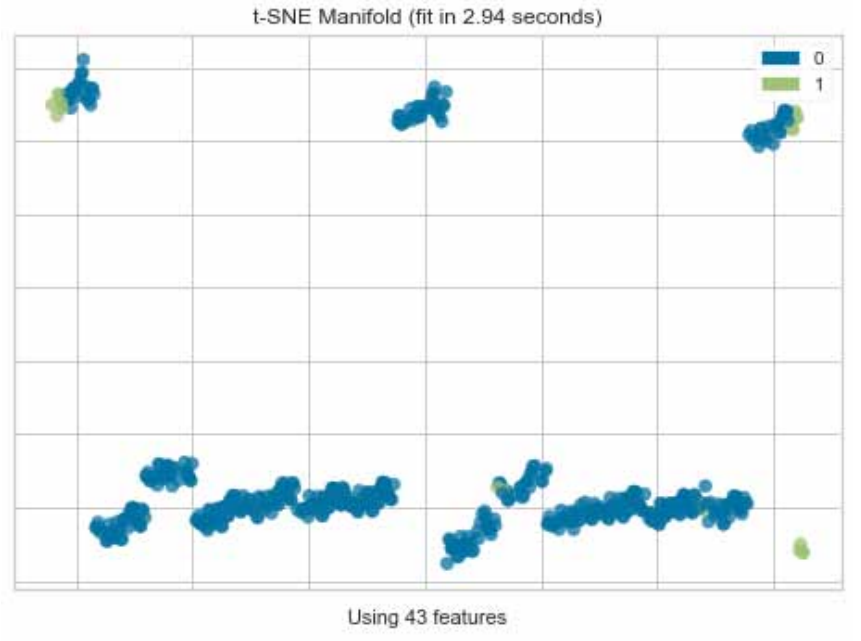


Figure 3(d). Applying Decision Tree (d) Calibration curve

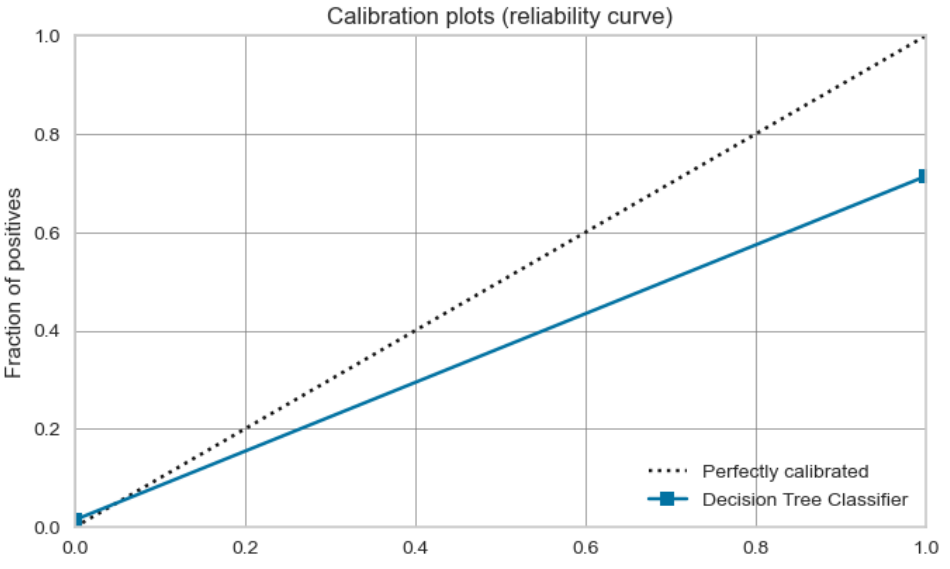


Figure 4(a). Applying LightGBM (a)ROC curves

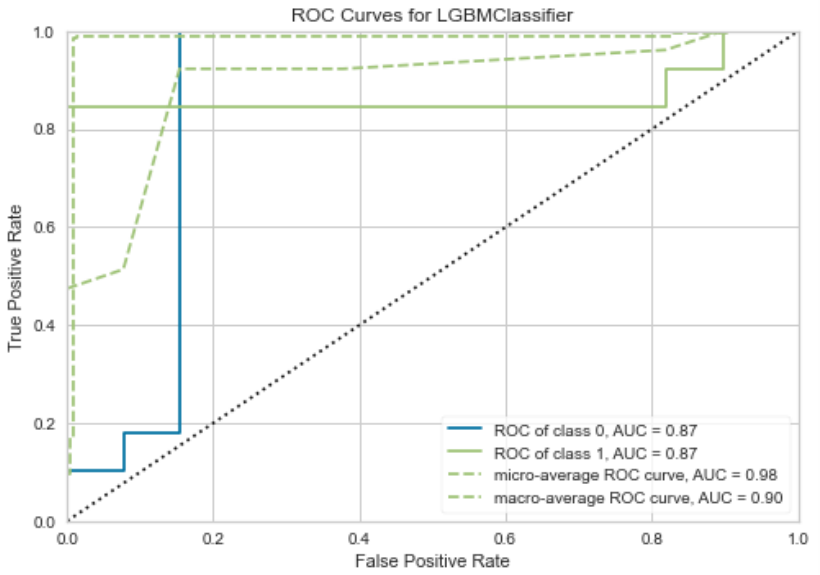


Figure 4(b). Applying LightGBM (b) Confusion matrix

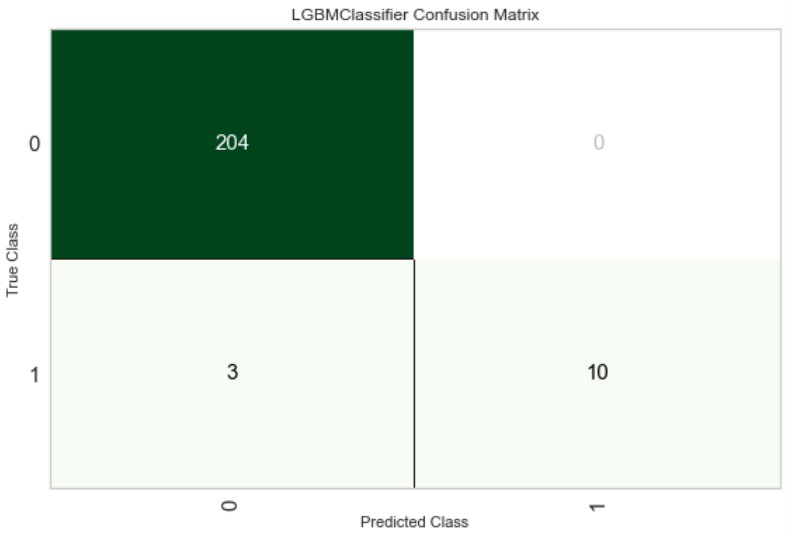


Figure 4(c). Applying LightGBM (c) t-sne manifold curvers

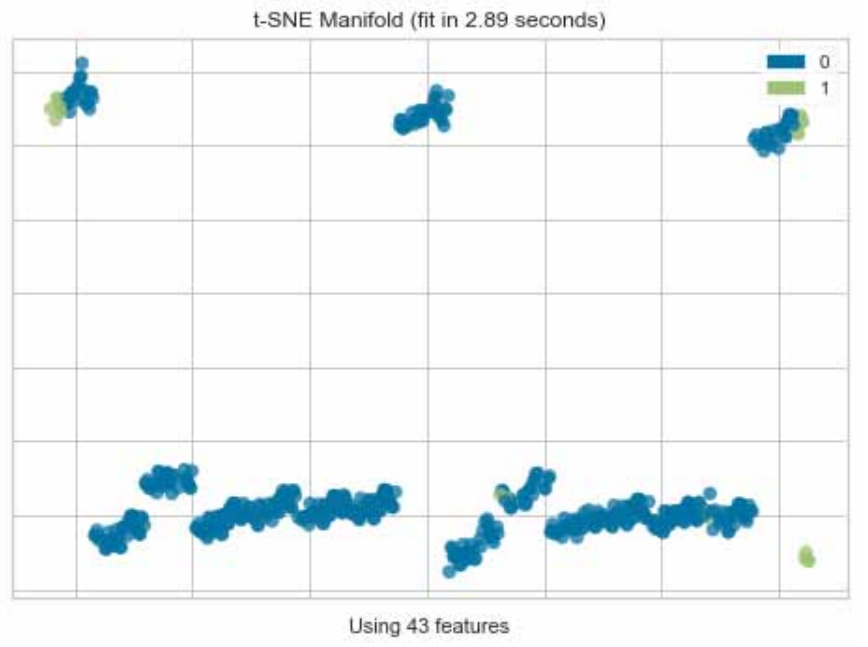


Figure 4(d). Applying LightGBM (d) Calibration curve

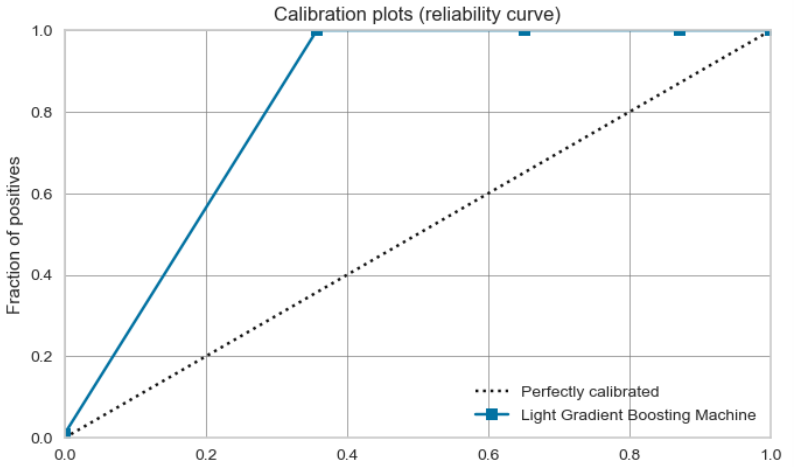


Figure 5(a). Applying KNN (a)ROC curves

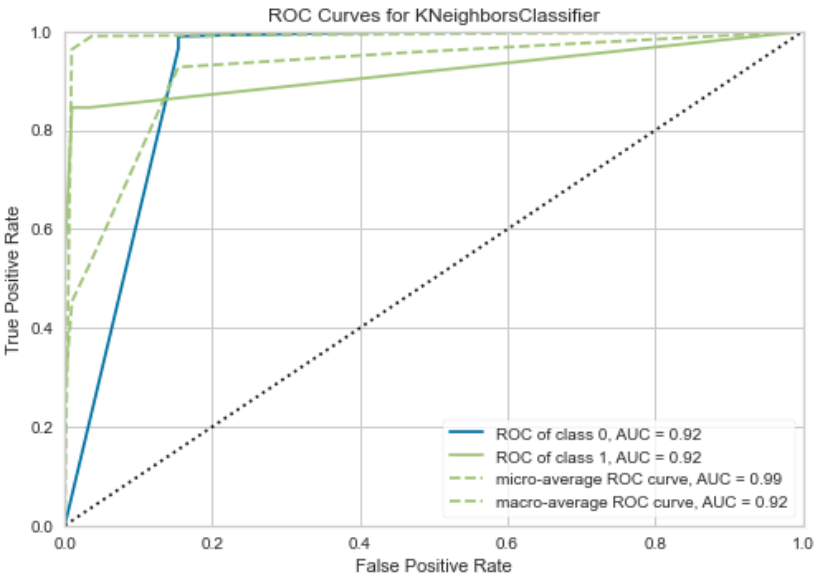


Figure 5(b). Applying KNN (b) Confusion matrix

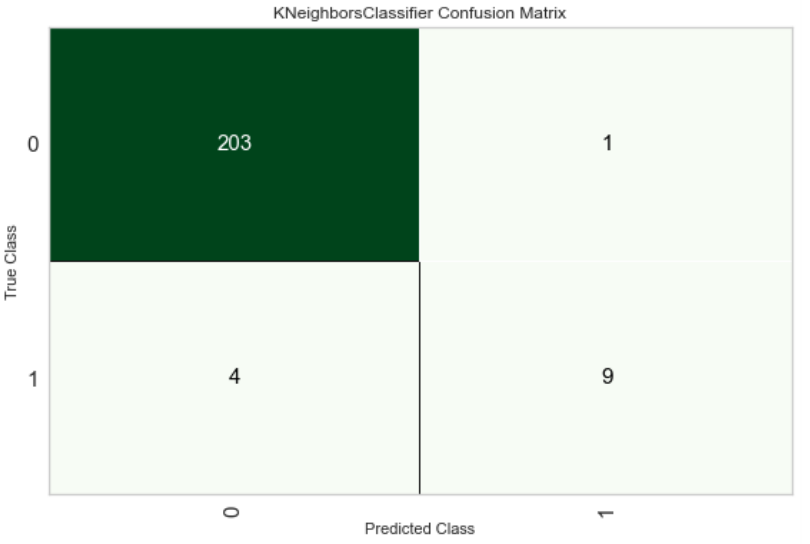


Figure 5(c). Applying KNN (c) t-sne manifold curves

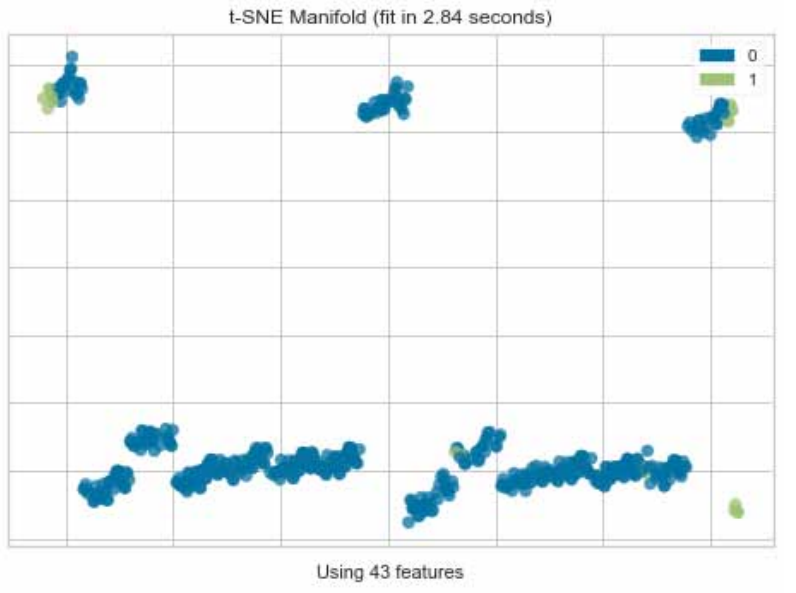


Figure 5(d). Applying KNN (d) Calibration curve

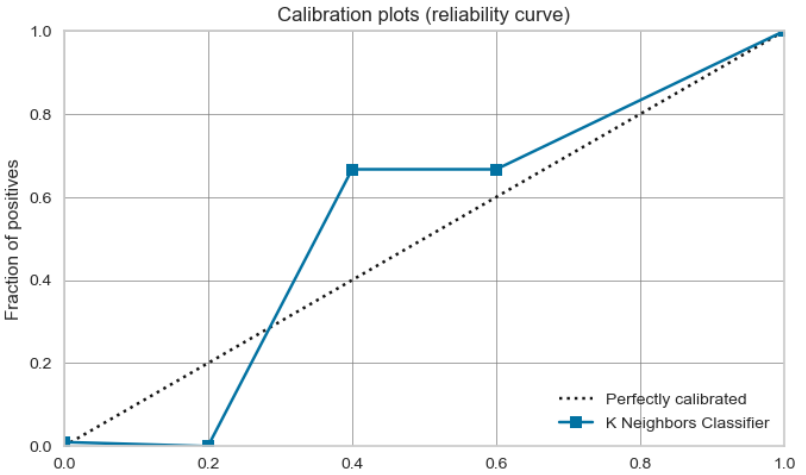


Figure 6(a). Applying LR (a) ROC curves

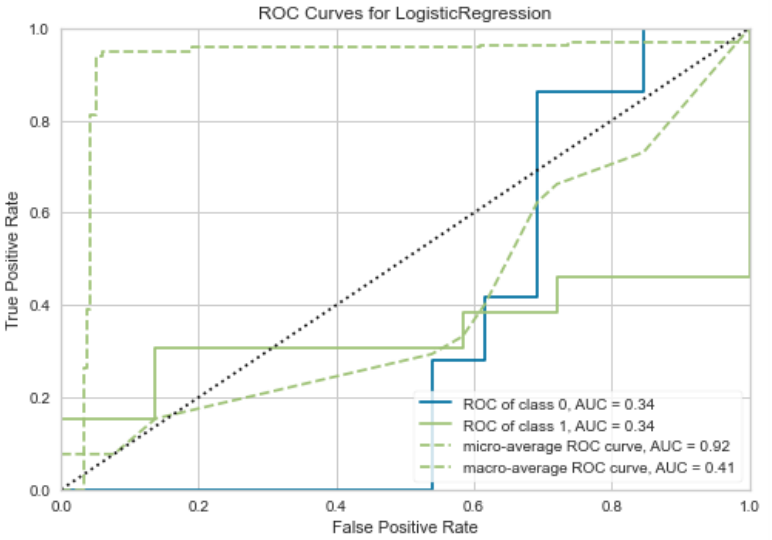


Figure 6(b). Applying LR (b) Confusion matrix

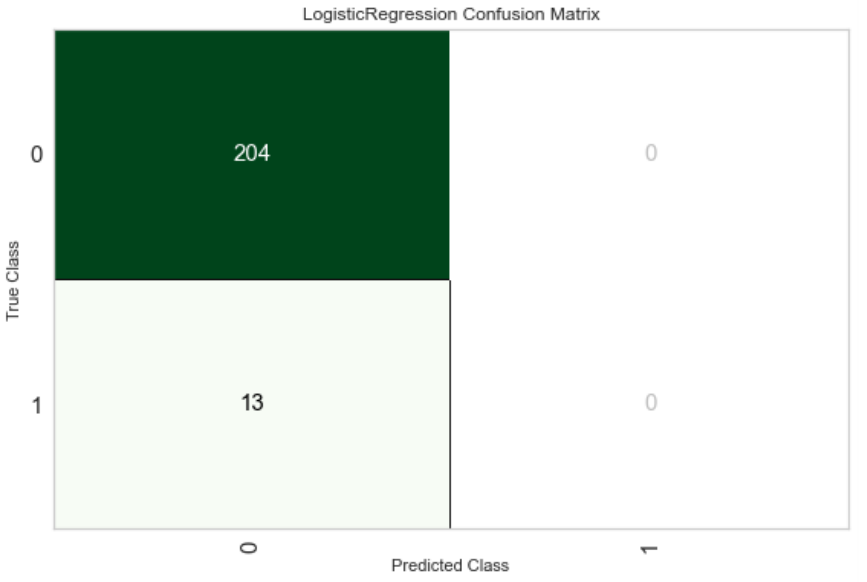


Figure 6(c). Applying LR (c) t-sne manifold curves

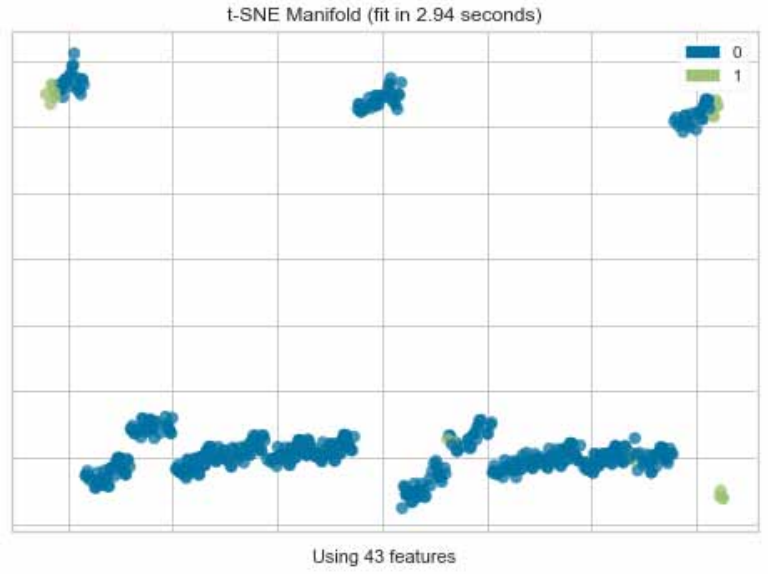
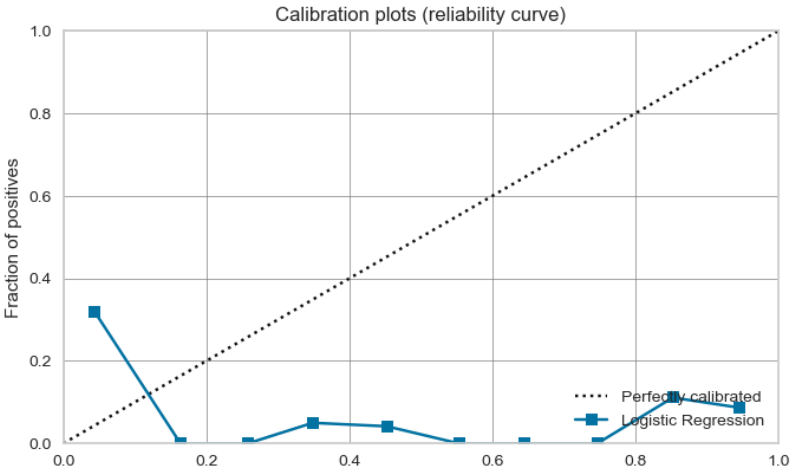


Figure 6(d). Applying LR (d) Calibration curve



For the Decision Tree algorithm, Fig 3 (a) The ROC for this algorithm shows that it is perfectly skilled and has an AUC of 0.87. There are 200 TP values, 4 FN values, 3 FP values, and 10 TN values as shown in Fig 3 (b). Fig 3 (d) shows the perfectly calibrated model where the Decision Tree classifier is a bit lower than the perfectly calibrated line, meaning the model fails at predicting each class label perfectly.

Using Random Forest, Fig 2 (a) ROC, the line starts from below left corner to the upper left corner and across the right upper corner. The larger values on the y-axis depict higher true positives and lower false negatives which clearly says the model to be perfectly skilled. In the confusion matrix, Fig 2 (b) shows the number of true positives is 203, the number of true negatives is 8 and the number of false positives and false negatives is 3 each. In Fig 2 (c) t-sne manifold graph shows the segregated groups of classes and it is used for dimensionality reduction.

Applying LightGBM, Fig 4 (a) depicts the curves for both positive(0) and negative(1) classes and it can be seen here that ROC for class 0 has high TPR than that of class 1. As shown in Fig 4 (b) It has an AUC of 0.98. Fig 4 (c) shows there are 204 True positives, 0 False negatives, 3 False positives, 10 True negatives which is a better result compared to other models.On KNN, there are 203 True Positives,1 False Negative,4 False positives, and 9 True negatives as seen in Fig 5 (b). Fig 5 (d) tells that it is far different from the perfectly calibrated line. Fig 5 (c) shows the divided groups where some groups contain positive and negative classes combined. Fig 5 (a) shows a good result.

All the above four algorithms gave good results unlike in logistic regression, where ROC is below the diagonal line meaning not perfectly skilled as seen in Fig 6 (a). Also, Fig 6 (b) shows 204 TP, 0 FN, 0 FP and 13 TN values. From the above results on this dataset, it can be seen that Decision Trees, Random Forest, LightGBM were good. But, considering RF has a high accuracy along with other metrics Precision, F1 score, Kappa, MCC having high values. Hence, Random Forest is considered the best for this dataset.

2) Semiconductor manufacturing process dataset

This dataset has a class label 'Pass/Fail' which tells that the manufacturing of semiconductors is successfully passed (-1) or failed (1). This means failure (1) indicates anomalies.

Table 4. Comparison of five classifiers on semiconductor manufacturing process dataset

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
knn	K Neighbors Classifier	0.9416	0.5104	0.0167	0.1000	0.0286	0.0274	0.0399	0.0260
lightgbm	Light Gradient Boosting Machine	0.9407	0.5083	0.0476	0.2500	0.0786	0.0712	0.0998	0.1530
rf	Random Forest Classifier	0.9398	0.4977	0.0333	0.2000	0.0571	0.0500	0.0724	0.2020
lr	Logistic Regression	0.9380	0.5039	0.0000	0.0000	0.0000	-0.0045	-0.0060	0.0310
dt	Decision Tree Classifier	0.8932	0.5093	0.0738	0.1117	0.0879	0.0323	0.0343	0.0180

Comparative analysis of five algorithms is given in Table4. KNN has given the best accuracy and AUC. Precision, Kappa, MCC were high for LightGBM and DT has a high score of Recall,F1.

Figure 7(a). Applying RF (a) ROC curves

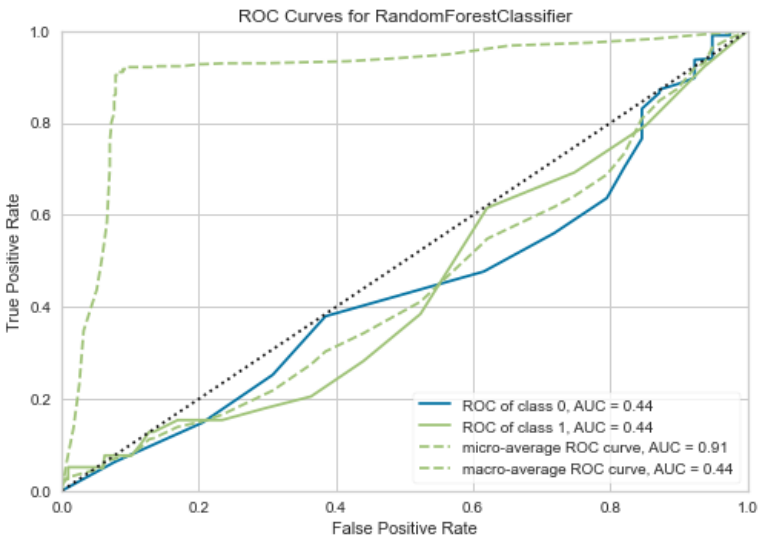


Figure 7(b). Applying RF (b) Confusion matrix

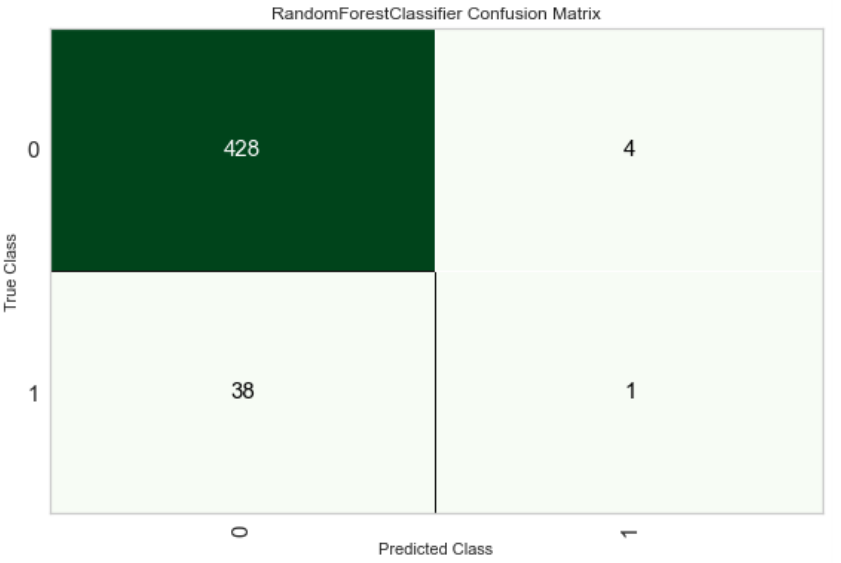


Figure 7(c). Applying RF (c) t-sne manifold curves

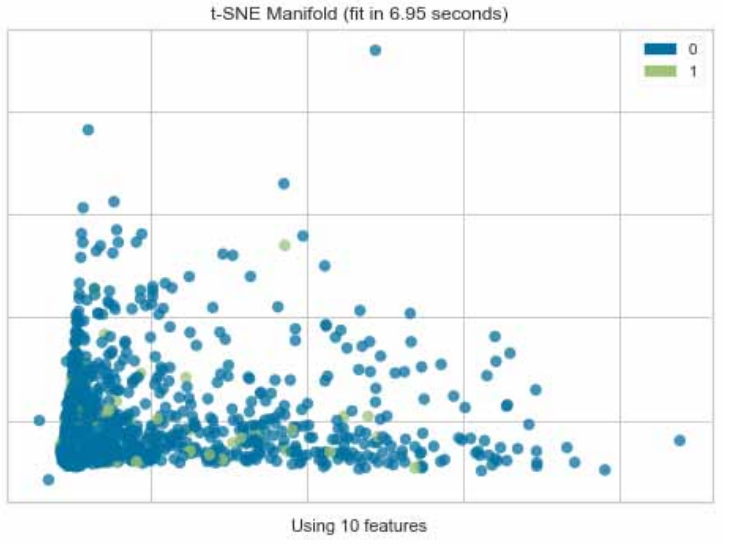


Figure 7(d). Applying RF (d) Calibration curve

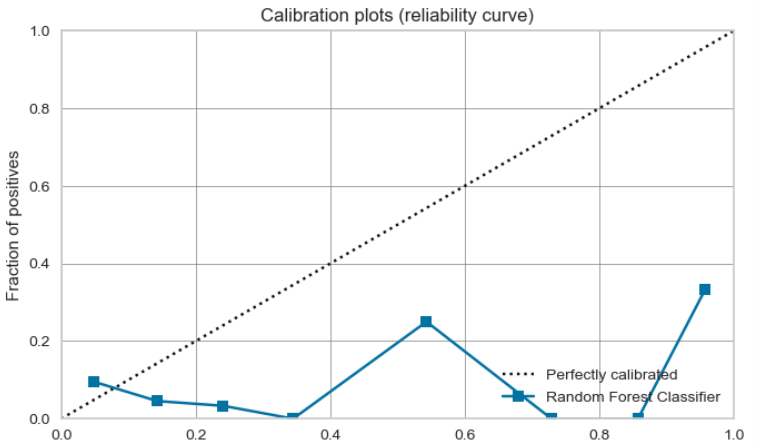


Figure 8(a). Applying DT (a) ROC curves

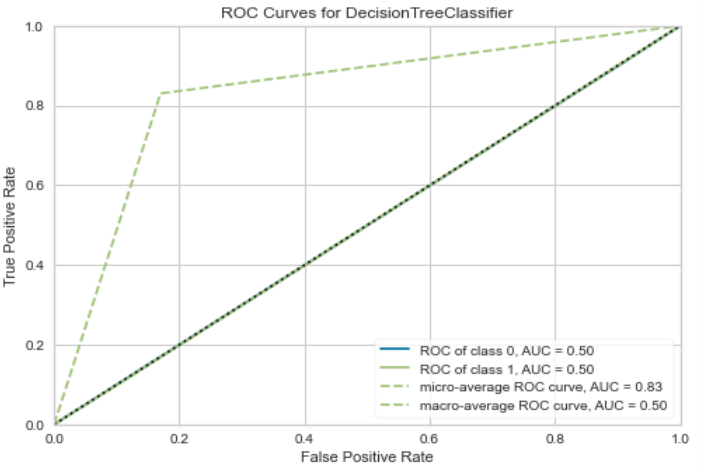


Figure 8(b). Applying DT (b) Confusion matrix

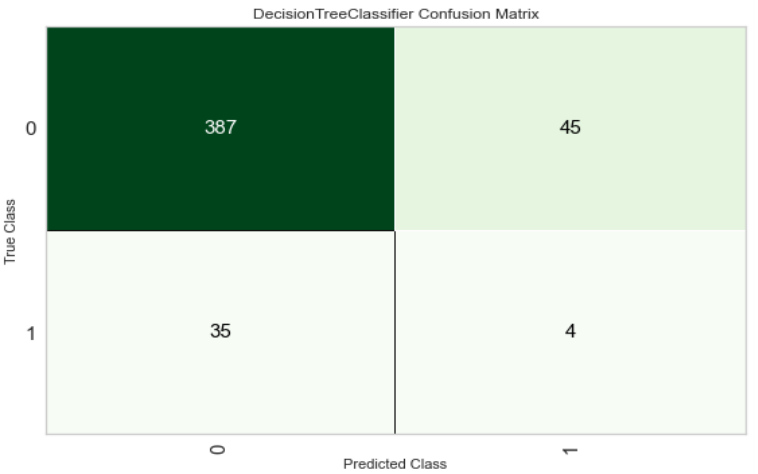


Figure 8(c). Applying DT (c) t-sne manifold curves

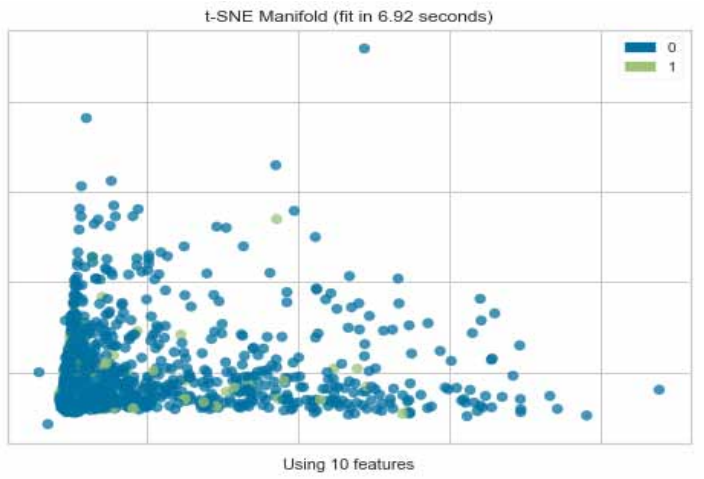


Figure 8(d). Applying DT (d) Calibration curve

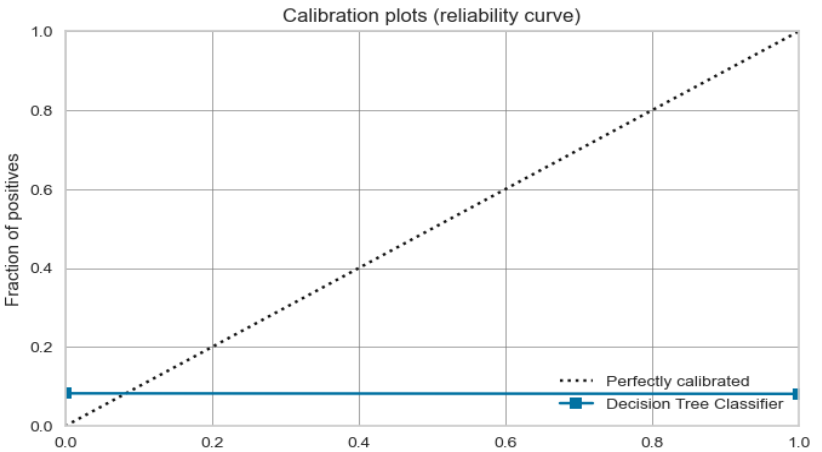


Figure 9(a). Applying LightGBM (a) ROC curves

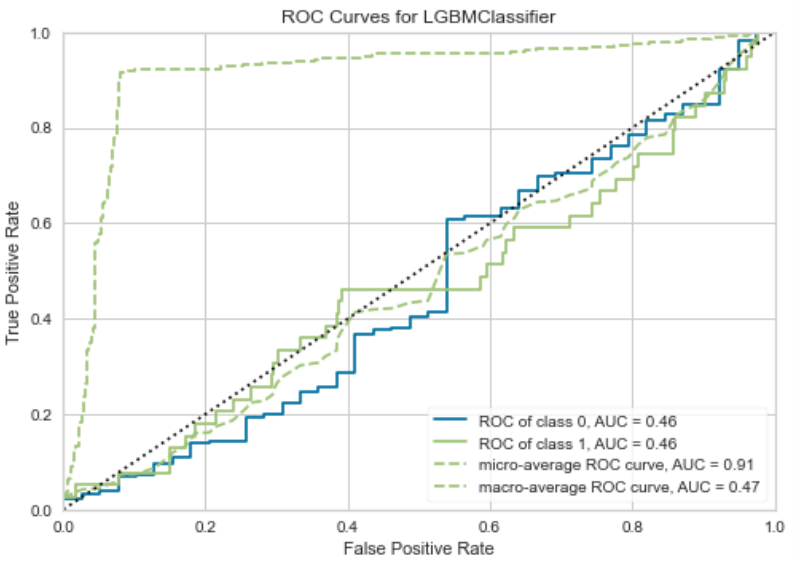


Figure 9(b). Applying LightGBM (b) Confusion matrix

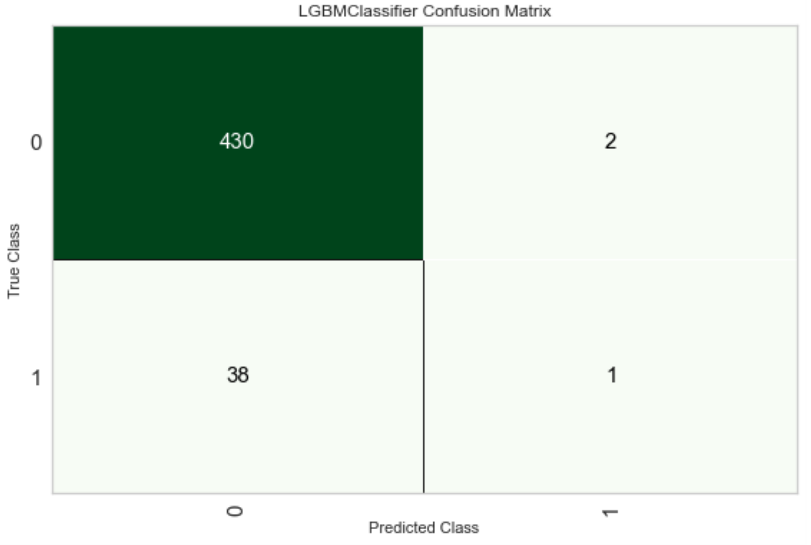


Figure 9(c). Applying LightGBM (c) t-sne manifold curves

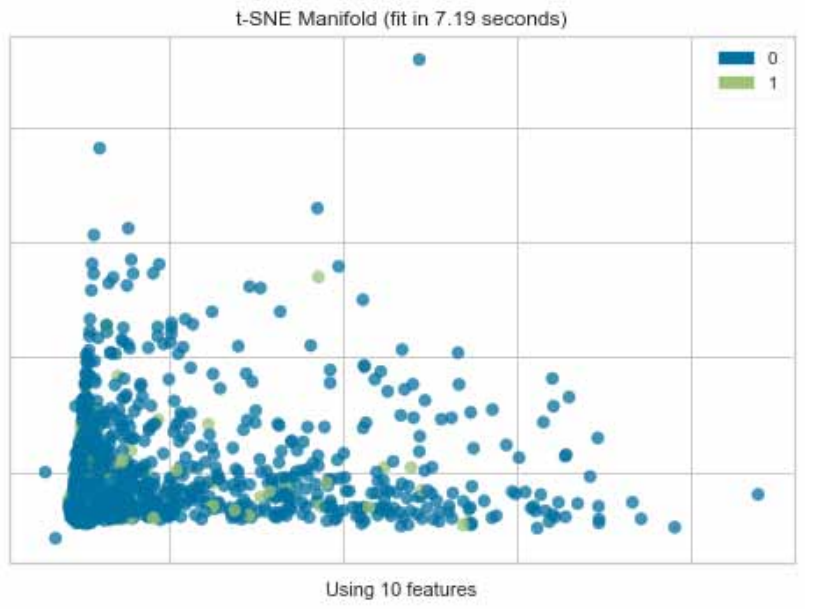


Figure 9(d). Applying LightGBM (d) Calibration curve

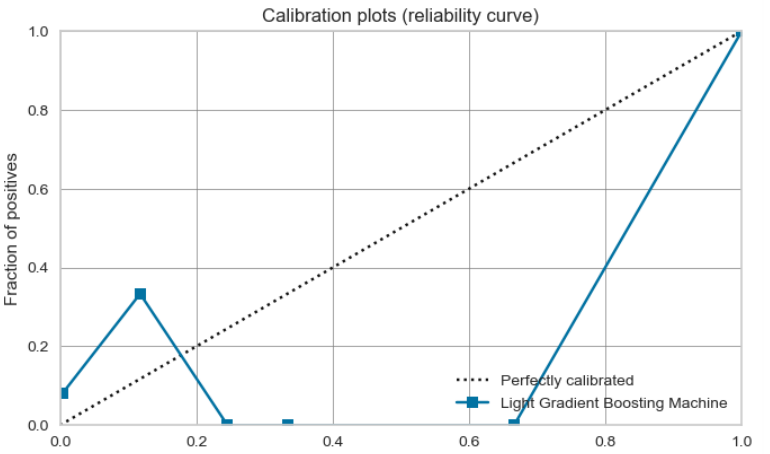


Figure 10(a). Applying KNN (a) ROC curves

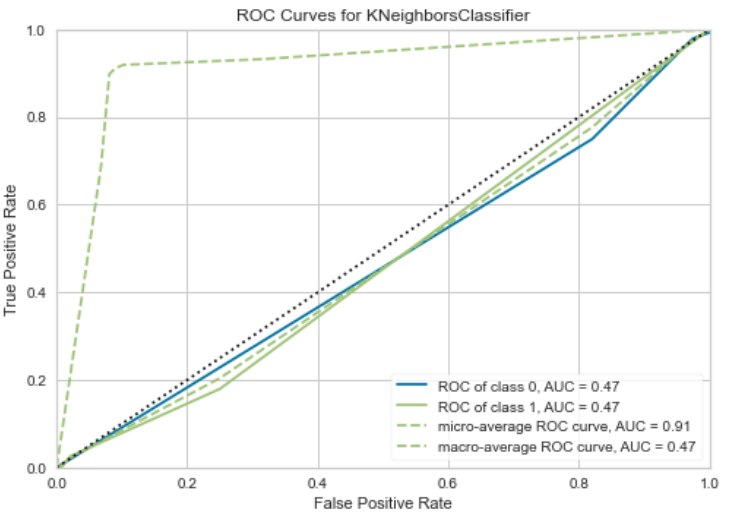


Figure 10(b). Applying KNN (b) Confusion matrix

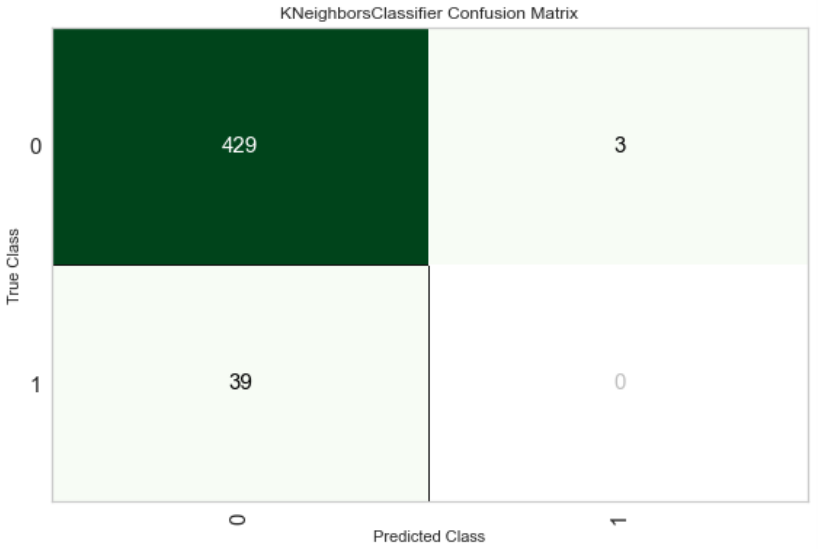


Figure 10(c). Applying KNN (c) t-sne manifold curves

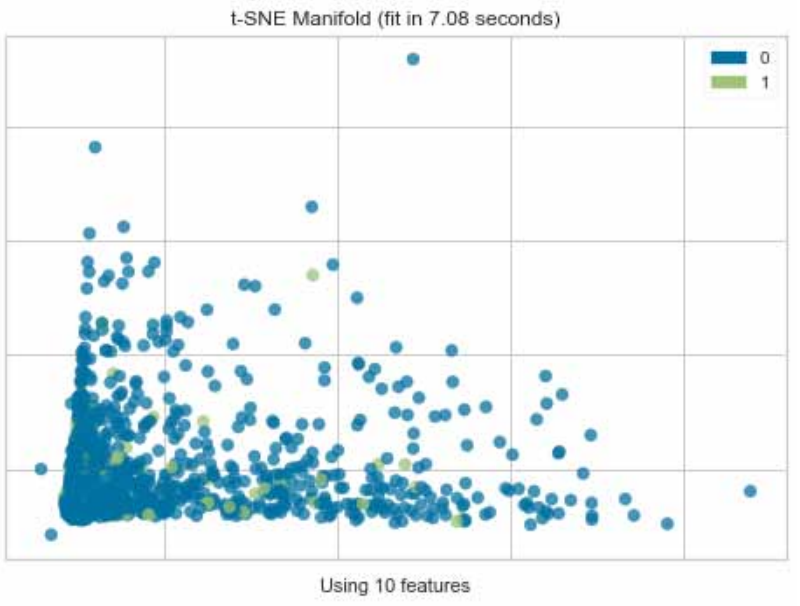


Figure 10(d). Applying KNN (d) Calibration curve

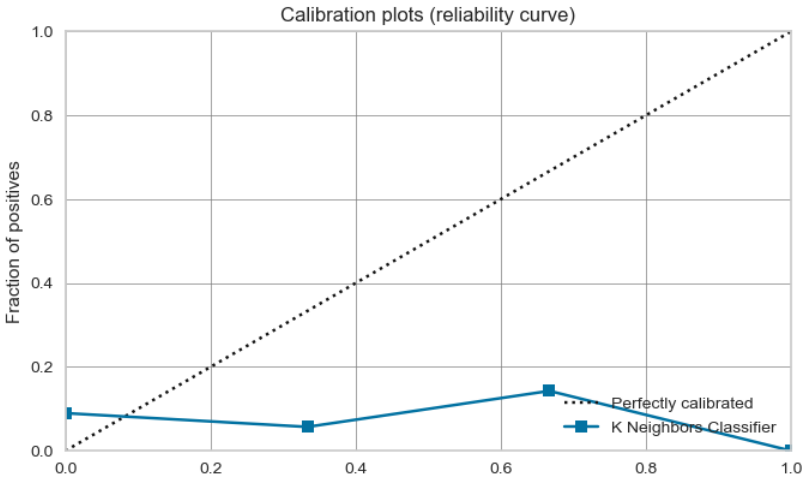


Figure 11(a). Applying LR (a) ROC curves

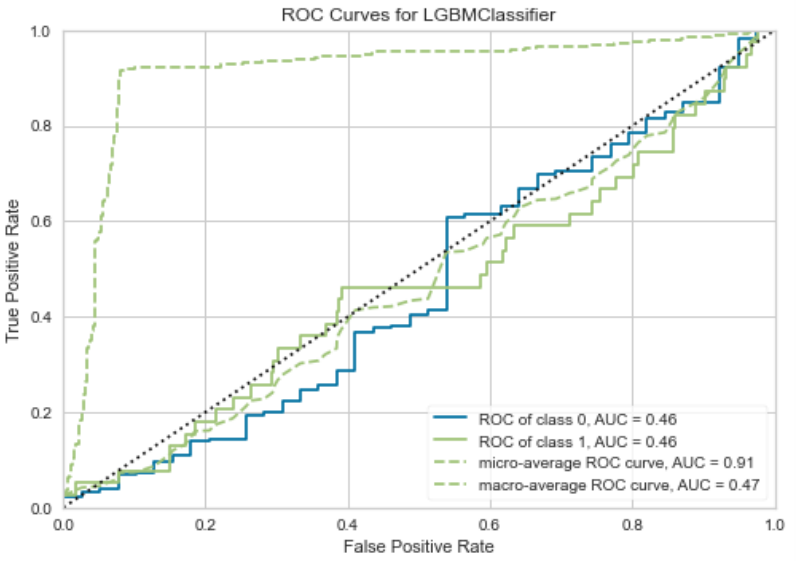


Figure 11(b). Applying LR (b) Confusion matrix

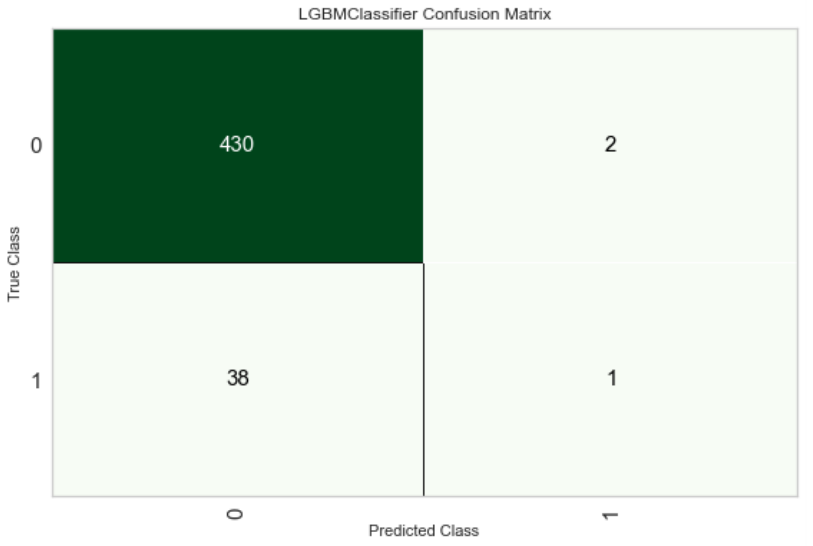


Figure 11(c). Applying LR (c) t-sne manifold curves

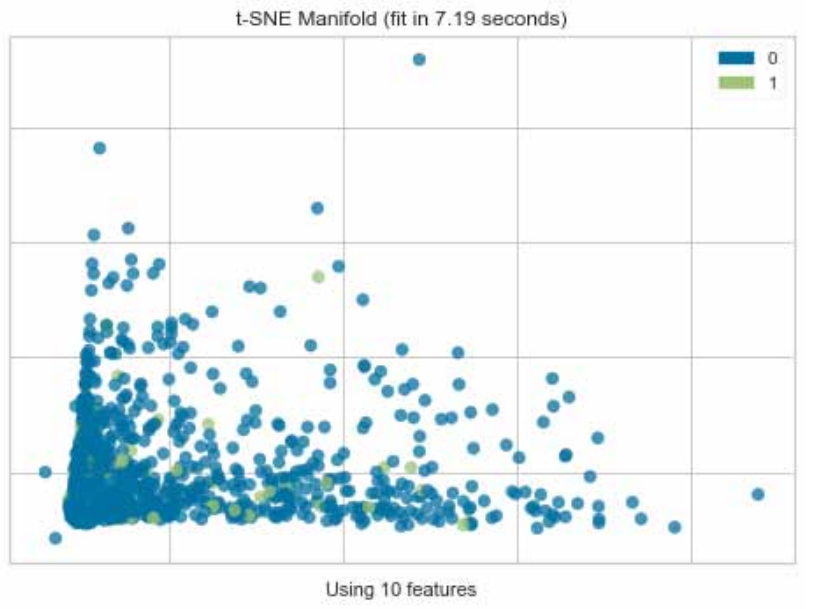
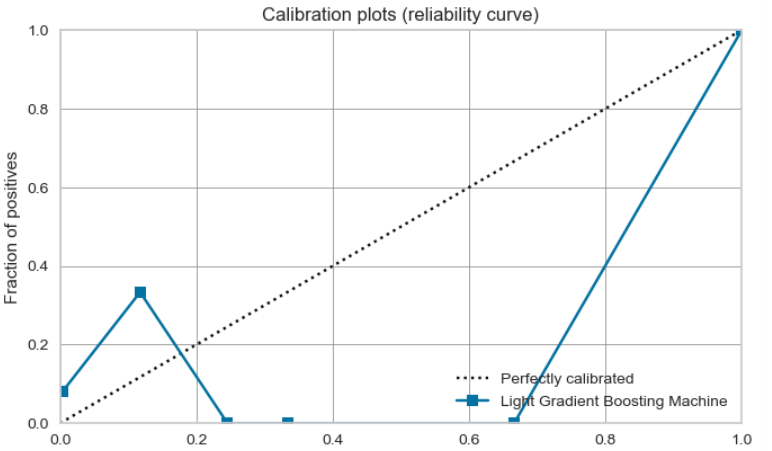


Figure 11(d). Applying LR (d) Calibration curve



For DT, Fig 8 (b) shows that there were 387 true positive values, 45 false negative values, 35 false-positive values, and 4 true negatives. This shows clearly that the error rate is high. Fig 8 (d) also shows that the model is not well calibrated.

Using RF, Fig 7 (a) is plotted below no skill line through micro average ROC is perfectly skilled. It has an AUC of 0.49 much less than the threshold as seen in the same. Regarding LightGBM, like ROC of RF, it gave the same curve. CM shows there were 430 TP, 2 FN, 38 FP, only 1 TN. The reliability curve shows that the model is not calibrated well.KNN gave good results compared to all other algorithms hence, it is the best model for this dataset.

3) DEMAND VS RESPONSE DATA FOR IOT ANALYTICS

For this dataset the Table 5 that shows the different algorithms performance metrics. RF and DT gave good results. But considering all the metrics RF can be concluded as the best model for this dataset.

The resultant plots are given below for each algorithm.

Table 5. Comparison of five classifiers on Demand vs Response Data for IoT analytics dataset

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
dt	Decision Tree Classifier	0.9999	0.9991	0.9982	1.0000	0.9991	0.9991	0.9991	0.0170
rf	Random Forest Classifier	0.9999	1.0000	0.9982	1.0000	0.9991	0.9991	0.9991	0.2730
lightgbm	Light Gradient Boosting Machine	0.9989	1.0000	0.9878	0.9896	0.9886	0.9880	0.9881	0.0620
knn	K Neighbors Classifier	0.9590	0.8485	0.2358	0.8155	0.3609	0.3466	0.4211	0.0440
lr	Logistic Regression	0.9519	0.6404	0.0486	0.2628	0.0759	0.0725	0.0960	0.0680

Figure 12(a). Applying RF (a) ROC curves

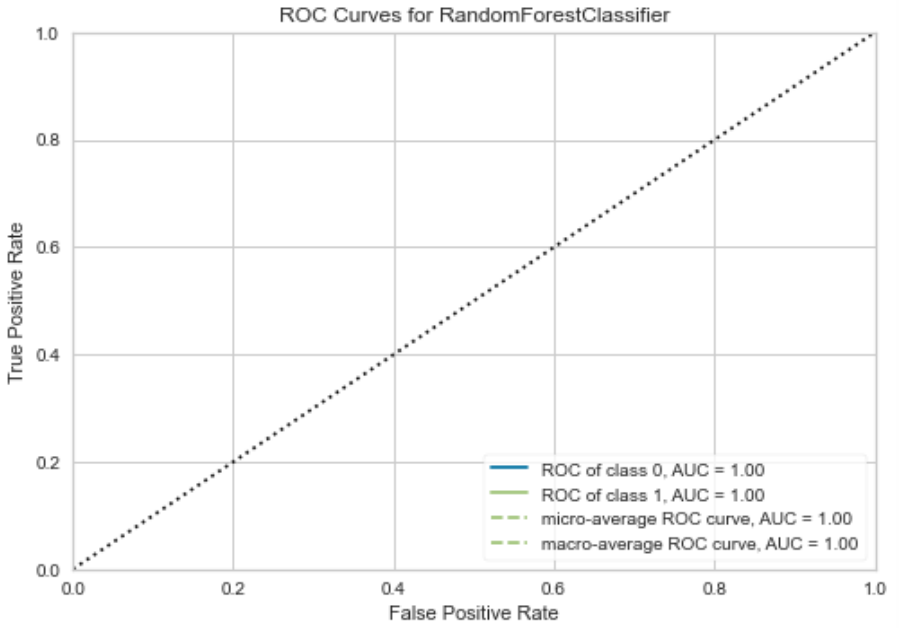


Figure 12(b). Applying RF (b) Confusion matrix

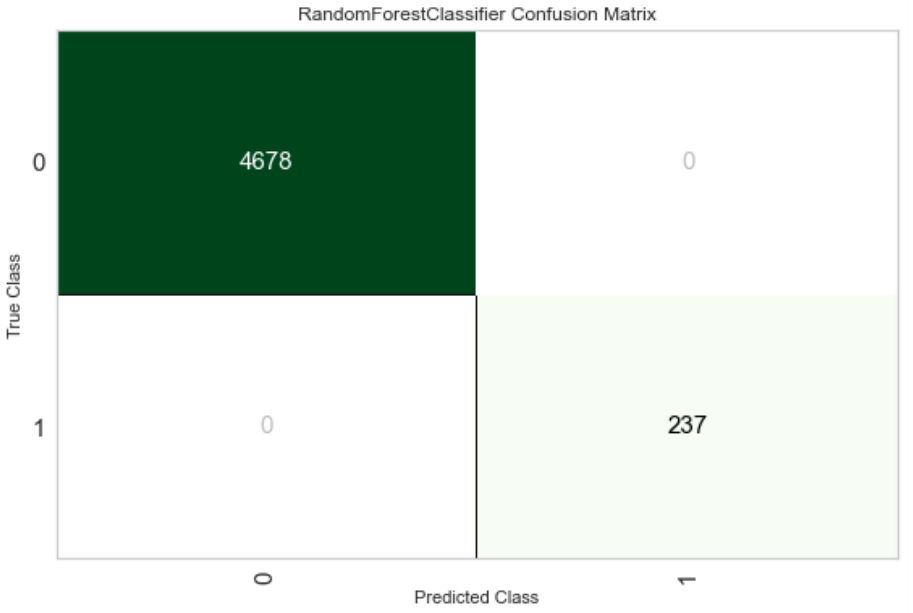


Figure 12(c). Applying RF (c) t-sne manifold curves

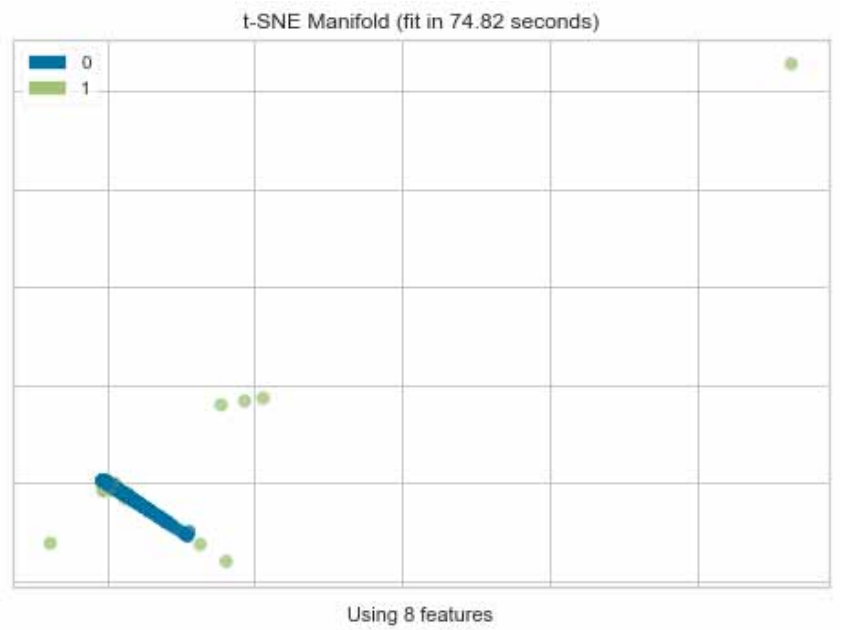


Figure 12(d). Applying RF (d) Calibration curve

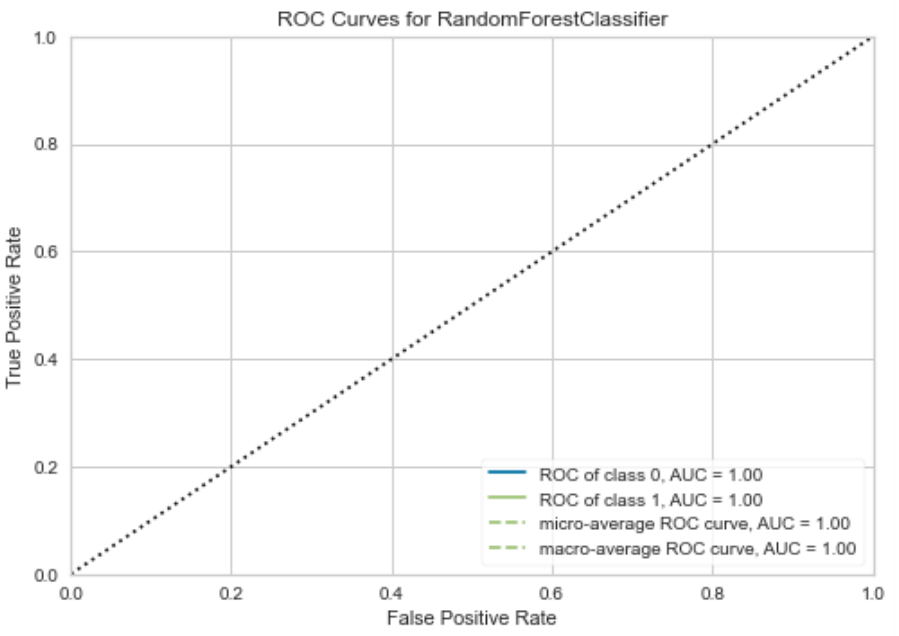


Figure 13(a). Applying DT (a) ROC curves

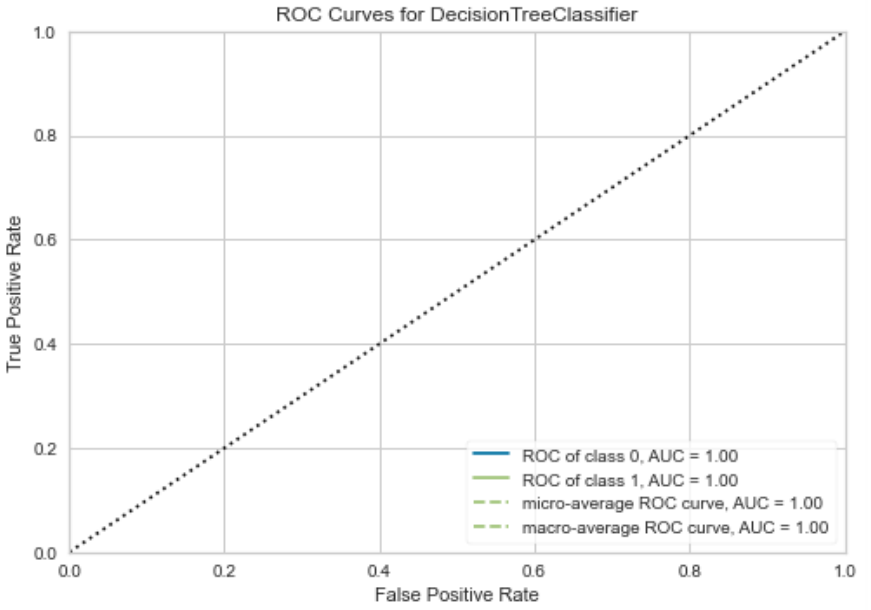


Figure 13(b). Applying DT (b) Confusion matrix

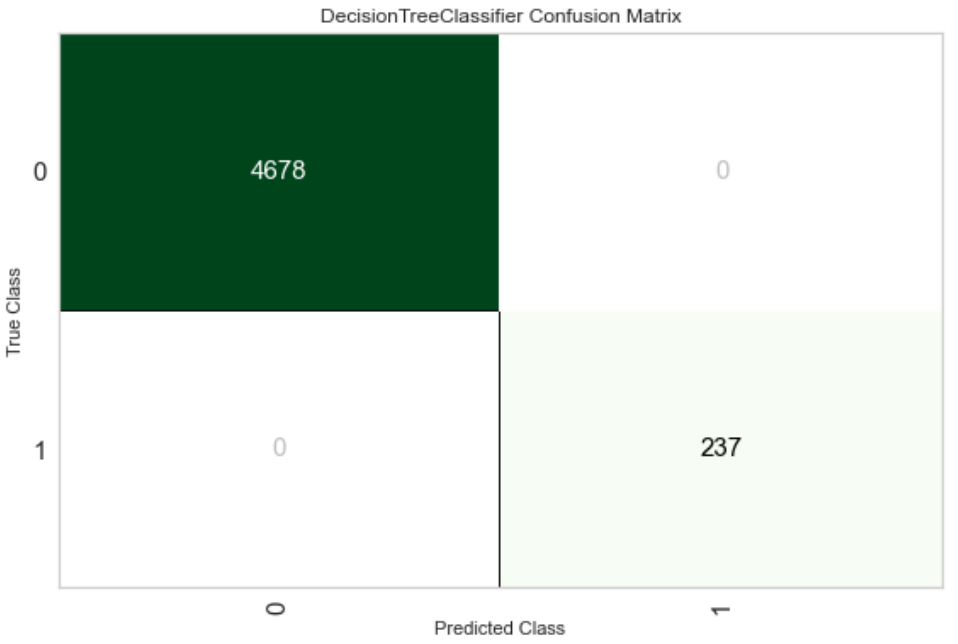


Figure 13(c). Applying DT (c) t-sne manifold curves

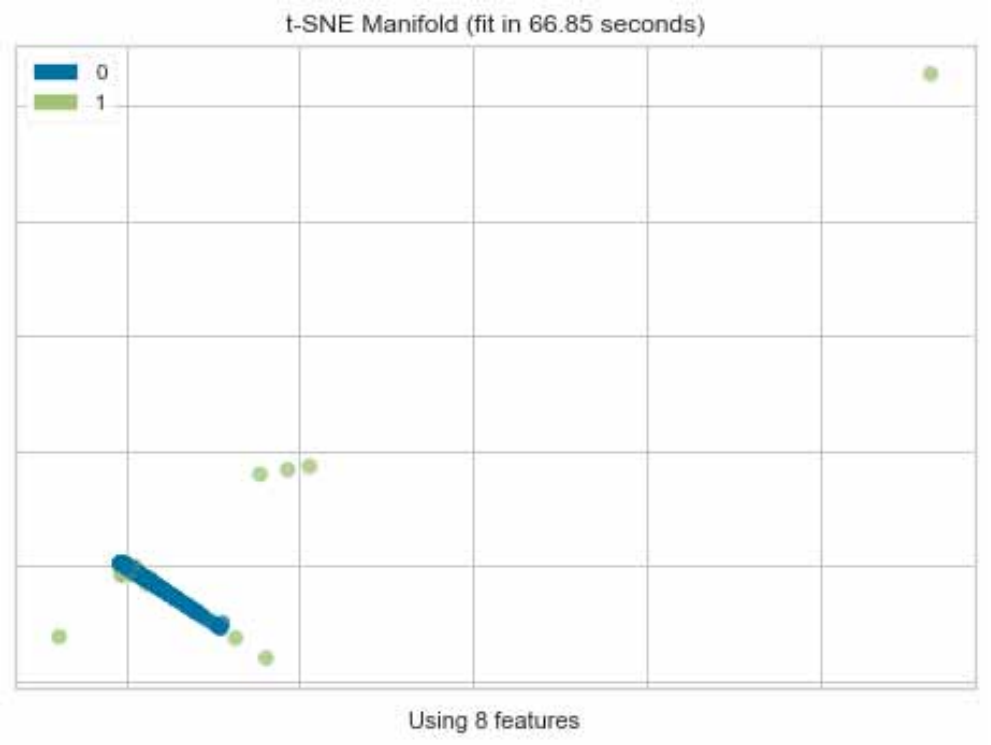


Figure 13(d). Applying DT (d) Calibration curve

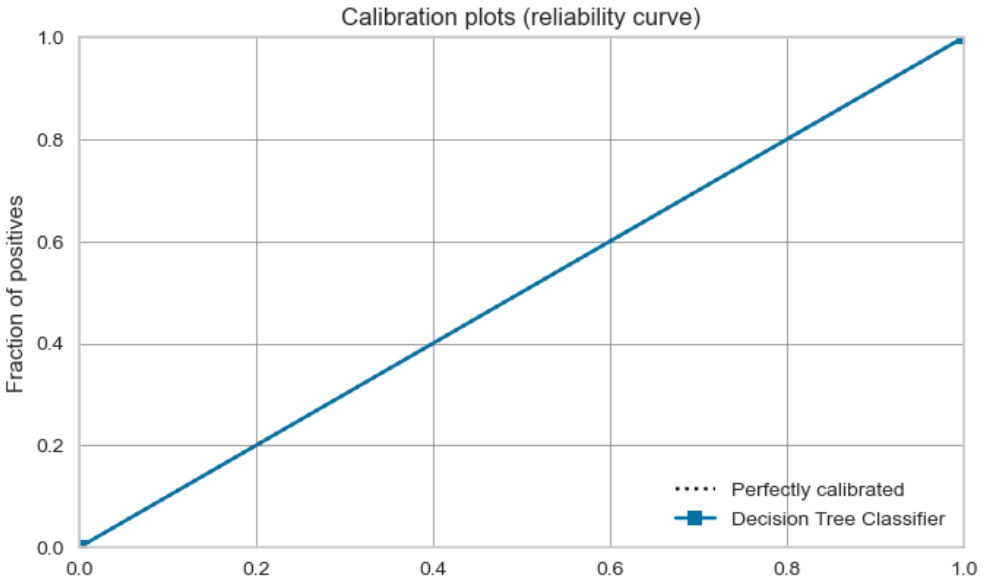


Figure 14(a). Applying LightGBM (a) ROC

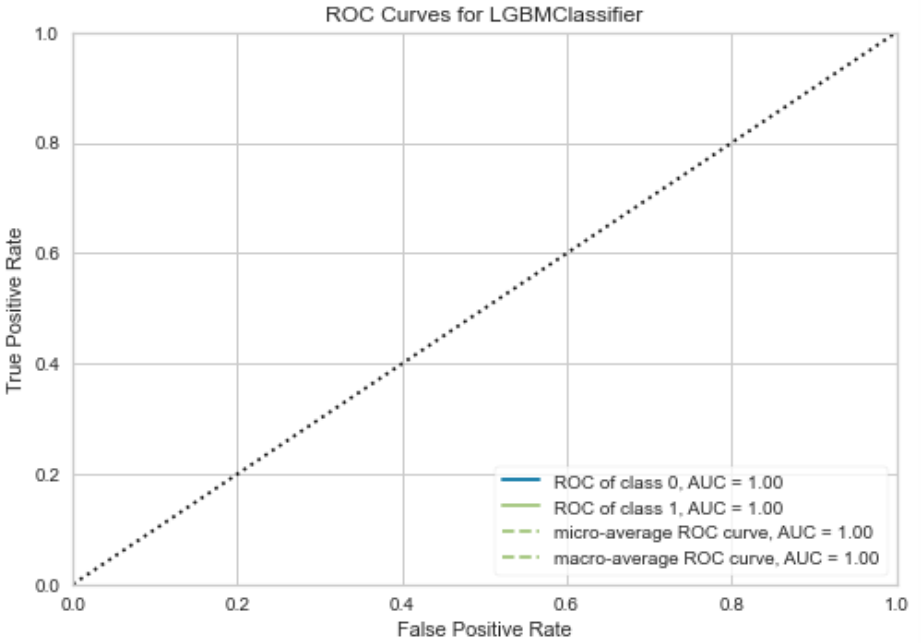


Figure 14(b). Applying LightGBM (b) Confusion matrix

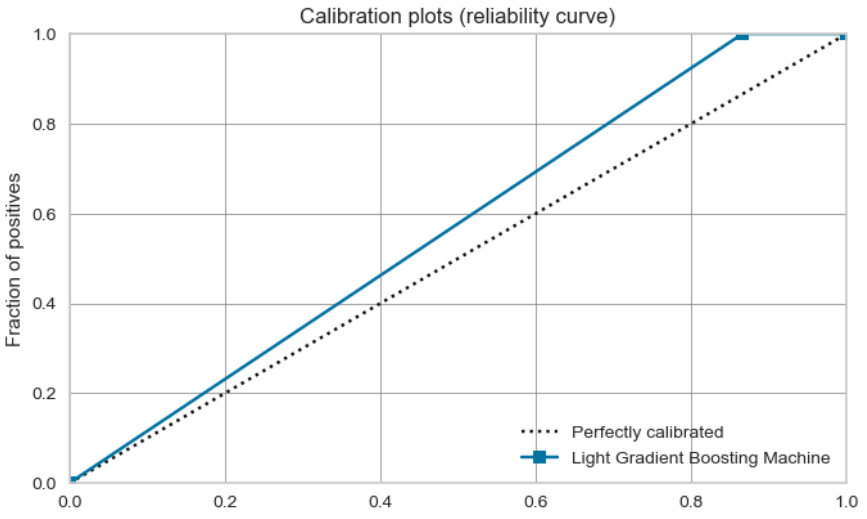


Figure 14(c). Applying LightGBM (c) t-sne manifold curves

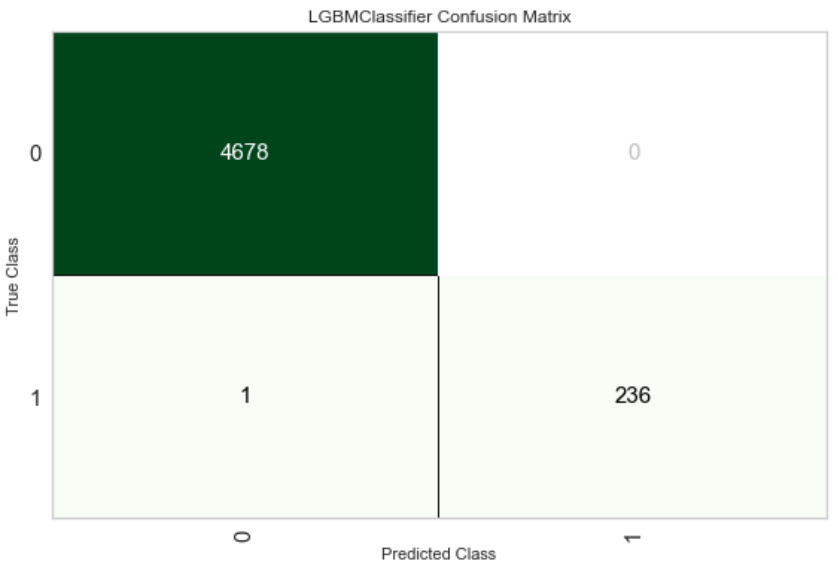


Figure 14(d). Applying LightGBM (d) Calibration curve

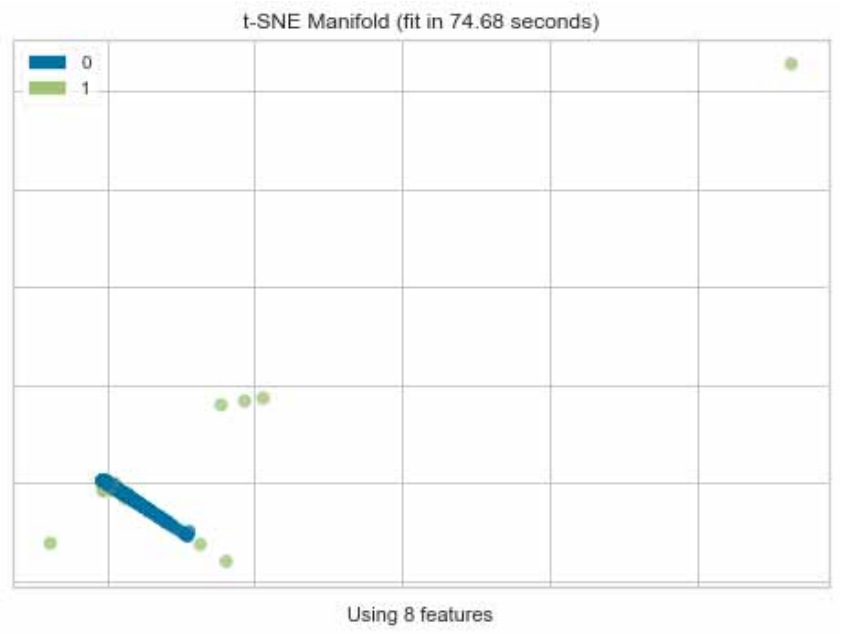


Figure 15(a). Applying KNN (a) ROC

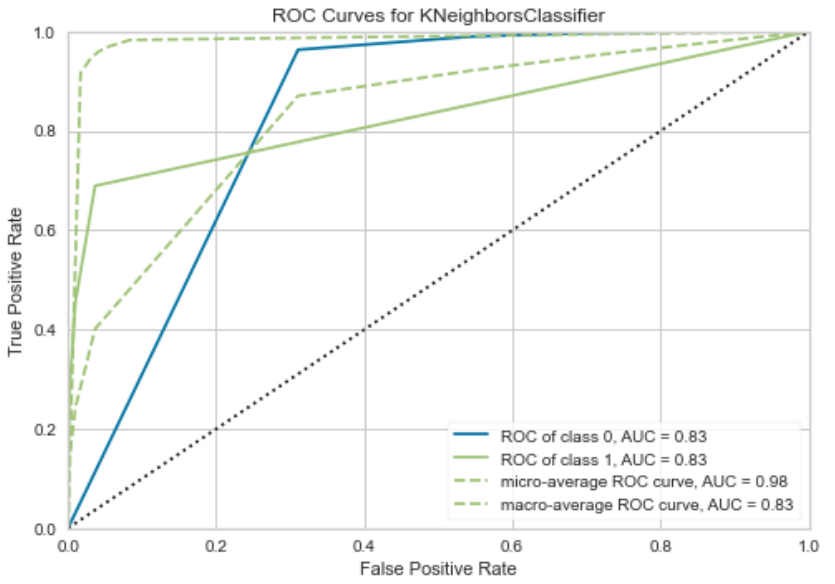


Figure 15(b) Applying KNN (b) Confusion matrix

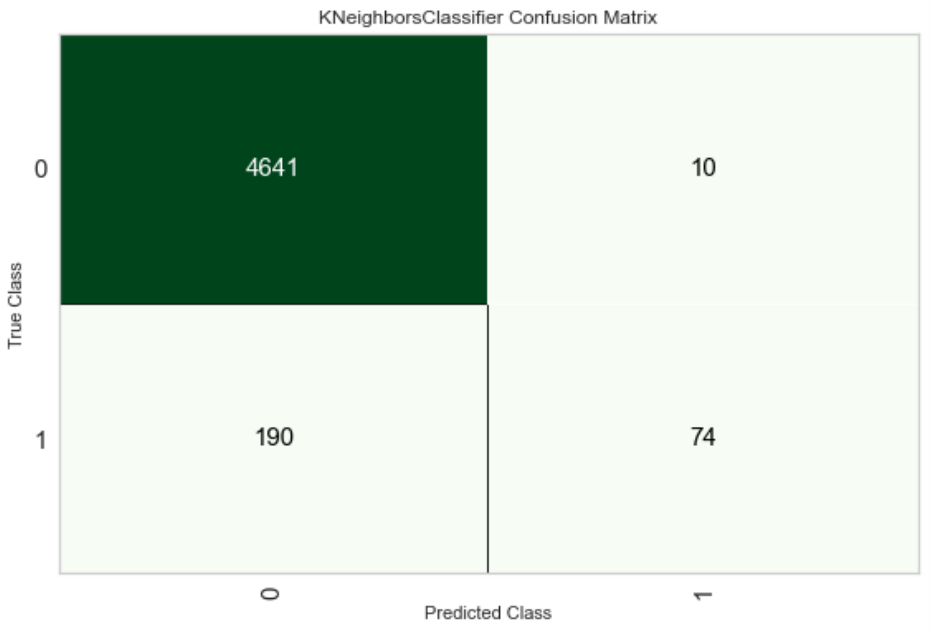


Figure 15(c). Applying KNN (c) t-sne manifold curves

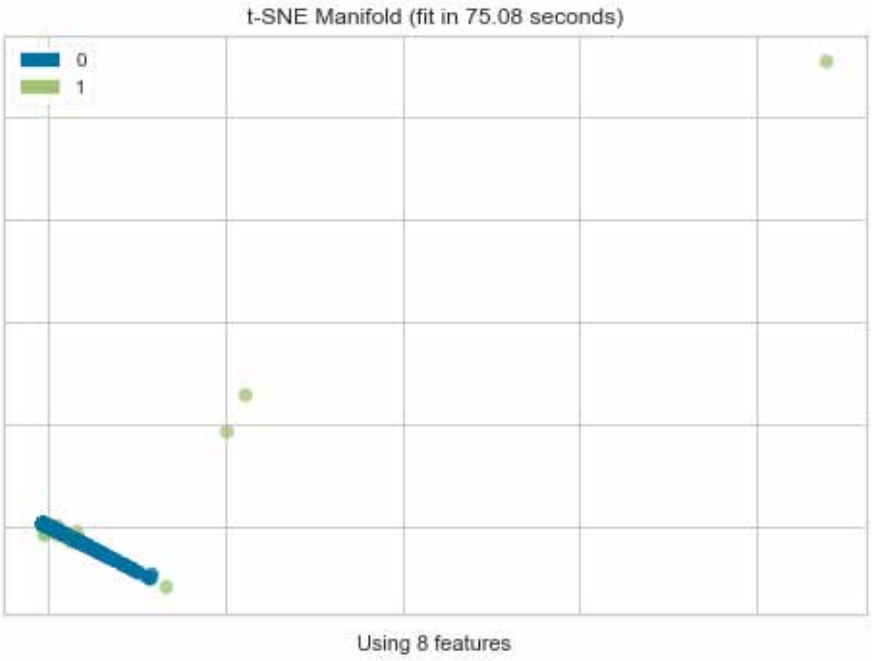


Figure 15(d). Applying KNN (d) Calibration curve

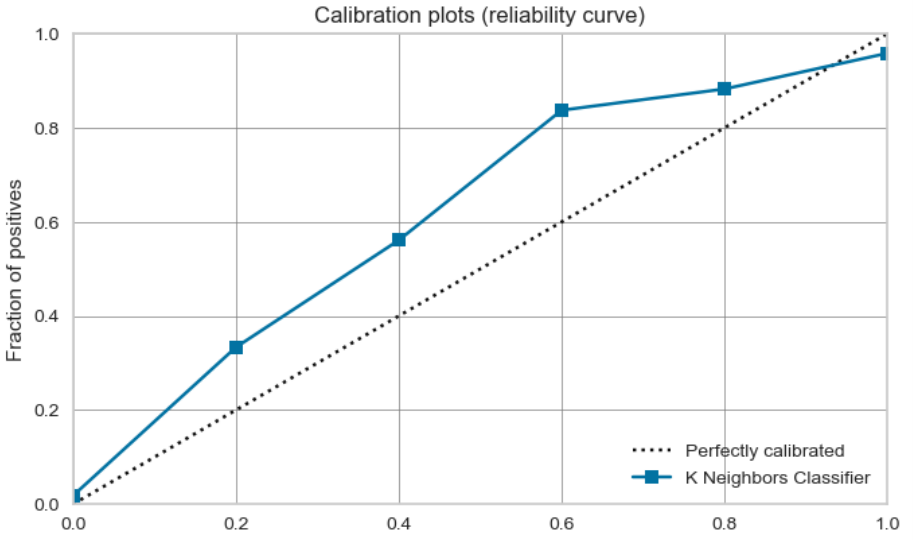


Figure 16(a). Applying LR (a) ROC curves

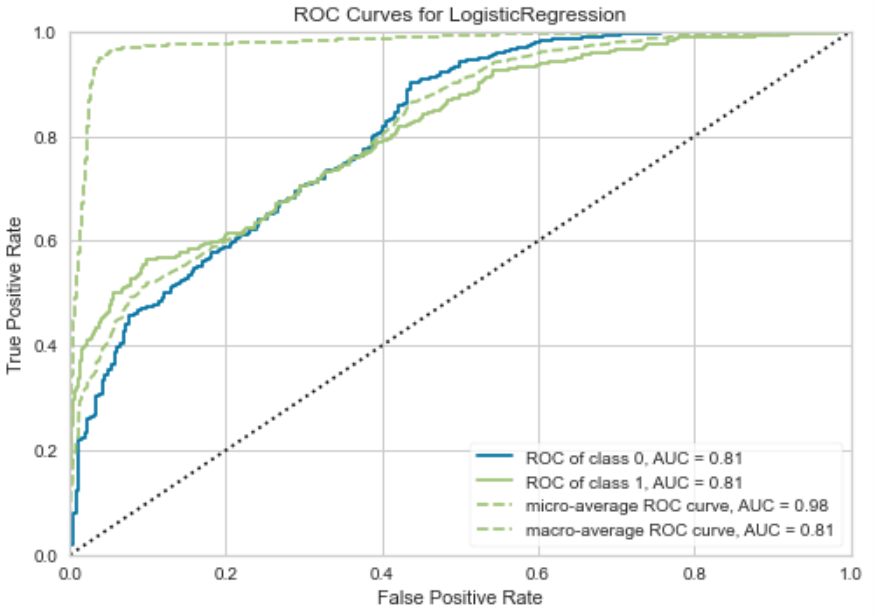


Figure 16(b). Applying LR (b) Confusion matrix

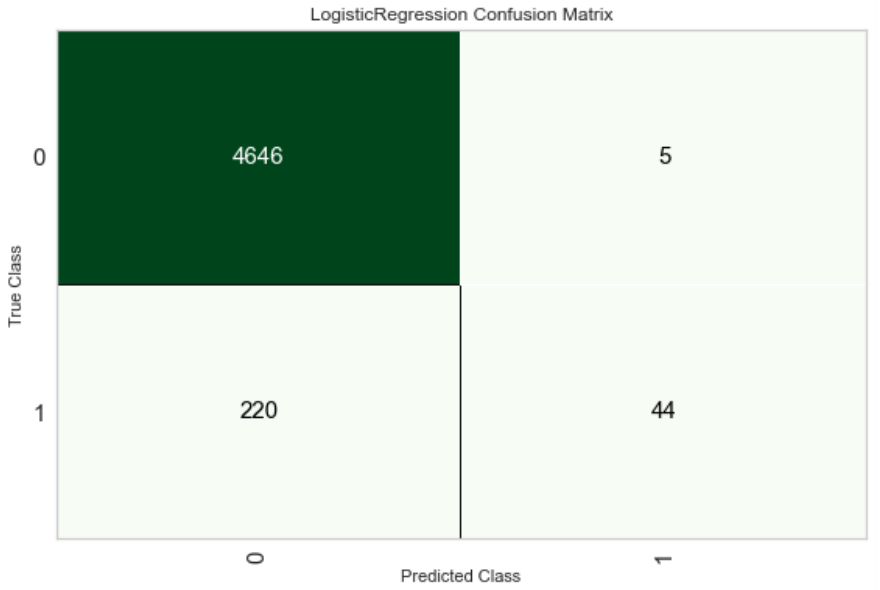


Figure 16(c). Applying LR (c) t-sne manifold curves

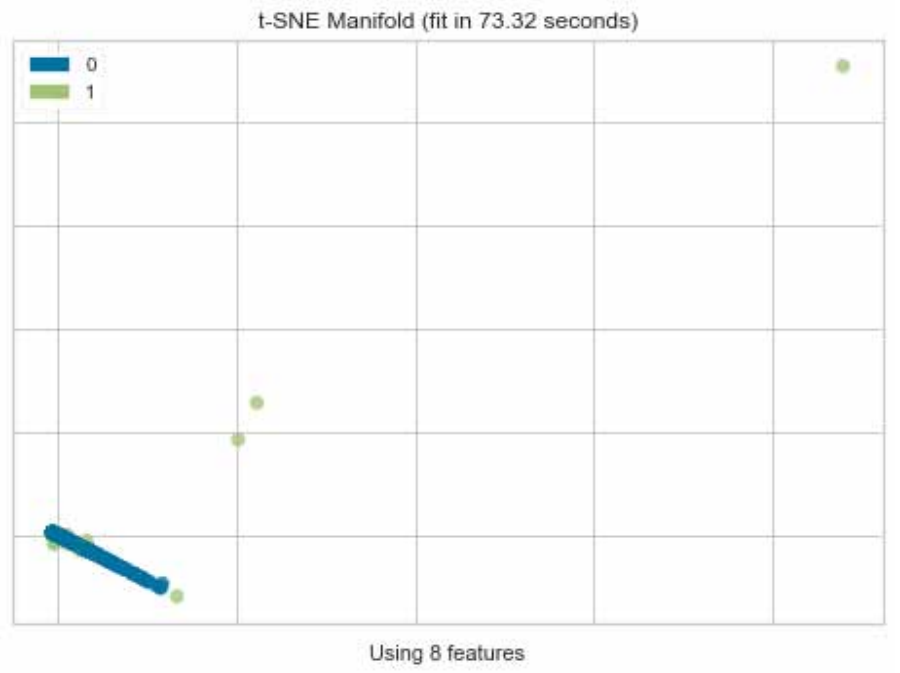
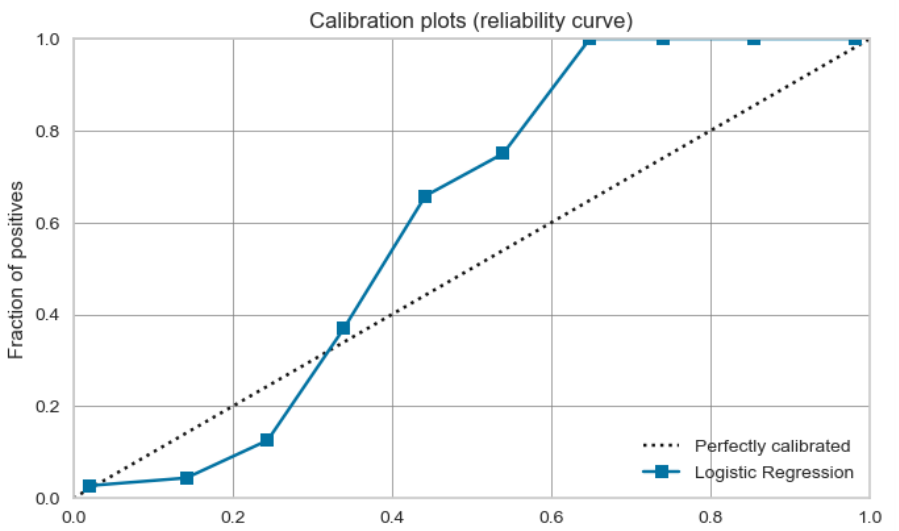


Figure 16(d). Applying LR (d) Calibration curve



For this dataset, considering DT, Fig 13 (a) shows that ROC is perfectly skilled with Fig 13 (b) showing 4678 True positives and 237 True negatives, 0 FP and FN, which means no errors. Fig 13 (d) shows the model is perfectly calibrated. The same is in the case of Random Forest. The confusion matrix in Fig 12 (b) shows there are no errors, 4678 True positives and 237 True negatives. In Fig 12 (a) ROC has got a perfectly skilled curve. LightGBM has also shown the same ROC as seen in Fig 14 (a) as DT and RF, though CM in Fig 14 (b) has shown one error with 4678 TP, 0 FN, 1 FP, and 235 TN.

Using the KNN model, Fig 15 (b) CM shows 4641 TP, 10 FN, 190 FP and 74 TN which tells the error is high and for the LR model also, in Fig 16 (b) the CM shows errors. Hence RF is considered the best model for this dataset though DT also showed good results, RF has given good outcomes for all the metrics.

4) HIGH STORAGE SYSTEM DATA FOR ENERGY OPTIMIZATION

Comparison of five classifiers for this dataset can be seen in Table 6 and the best model is determined to be Random Forest considering all the metrics.

The resultant plots are given below for each algorithm.

Table 6. Comparison of five classifiers on HRSS_anomalous_standard data

	Model	Accuracy	AUC	Recall	Prec.	F1	Kappa	MCC	TT (Sec)
rf	Random Forest Classifier	0.9815	0.9948	0.9422	0.9792	0.9603	0.9482	0.9485	2.854
dt	Decision Tree Classifier	0.9651	0.9510	0.9242	0.9291	0.9265	0.9036	0.9037	0.482
lightgbm	Light Gradient Boosting Machine	0.9460	0.9813	0.7892	0.9802	0.8740	0.8402	0.8485	0.413
knn	K Neighbors Classifier	0.8144	0.8720	0.5306	0.6319	0.5765	0.4589	0.4619	0.207
lr	Logistic Regression	0.7592	0.6539	0.0236	0.4054	0.0445	0.0190	0.0465	1.365

For this dataset, considering RF, it is seen in Fig 17(a) that ROC is perfectly skilled, Fig 17(b) depicts CM with 5338 true positives, 29 false negatives, 84 false positives and 1643 true negatives. Fig 17(c) shows t-sne manifold curves using 17 features.

Figure 17(a). Applying RF (a) ROC curves

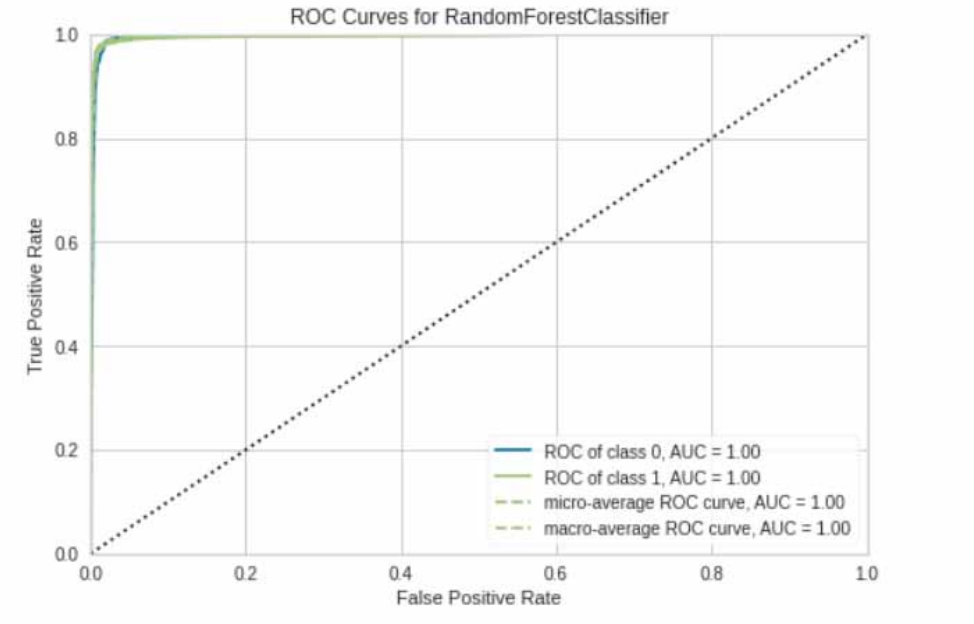


Figure 17(b). Applying RF (b) Confusion matrix

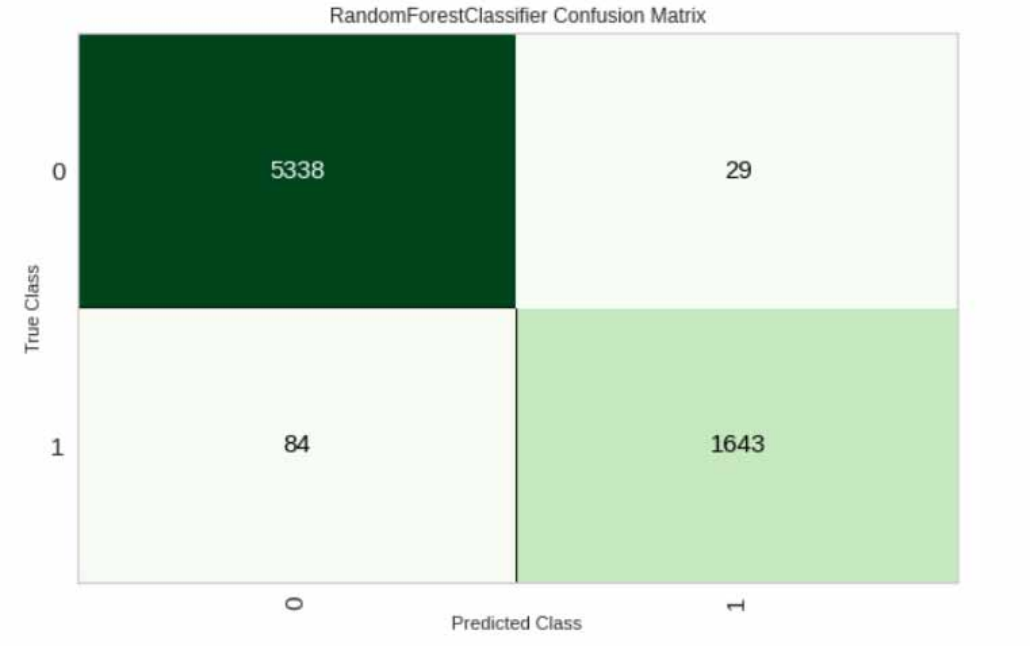


Figure 17(c) Applying RF (c) t-sne manifold curves

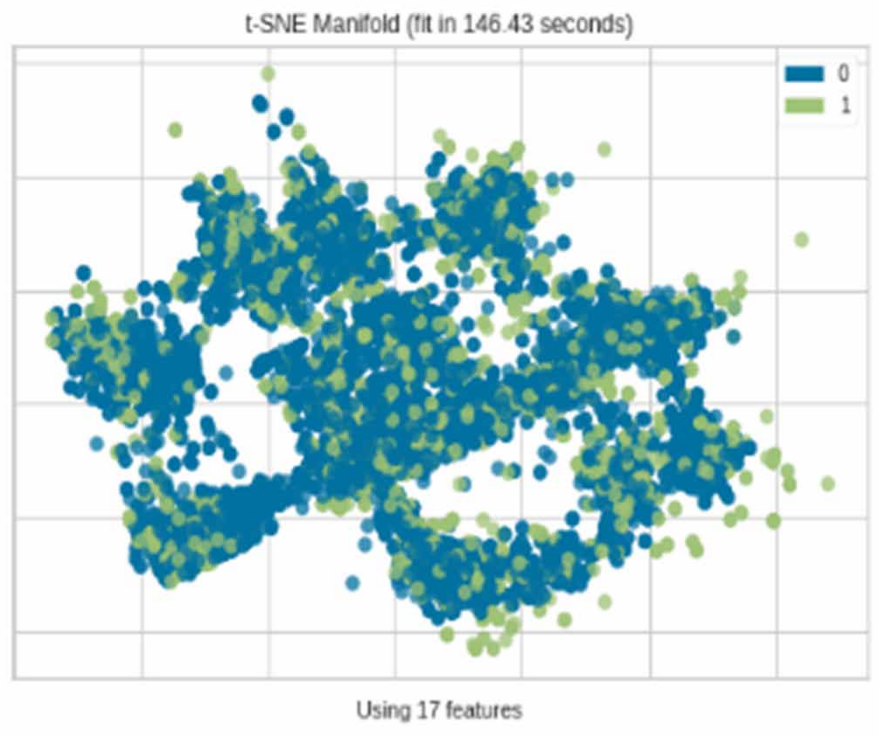


Figure 17(d). Applying RF (d) Calibration curve

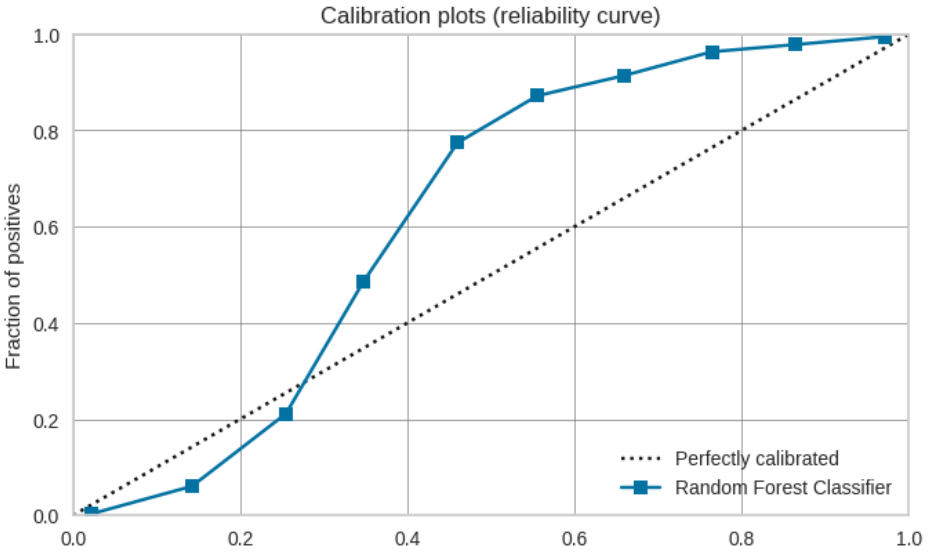


Figure 18(a). Applying DT (a) ROC curves

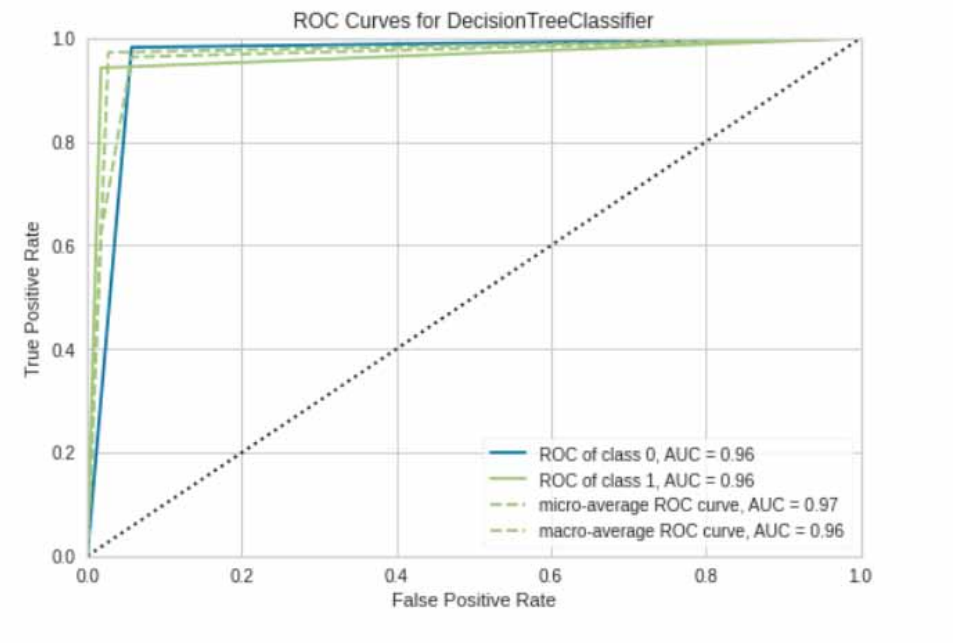


Figure 18(b). Applying DT (b) Confusion matrix

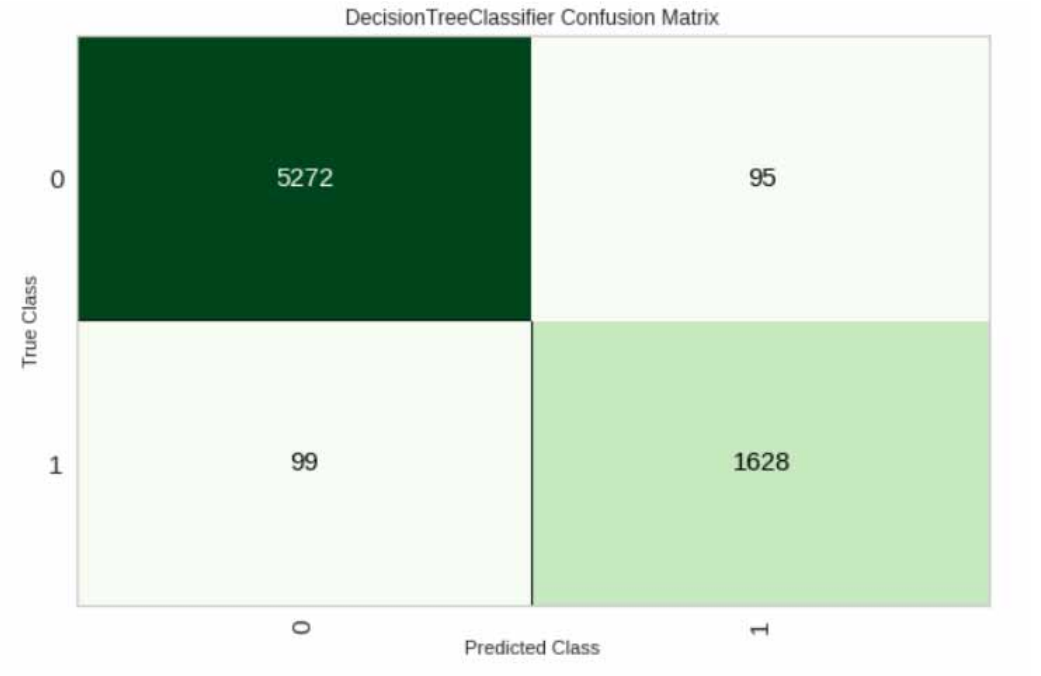


Figure 18(c) Applying DT (c) t-sne manifold curves

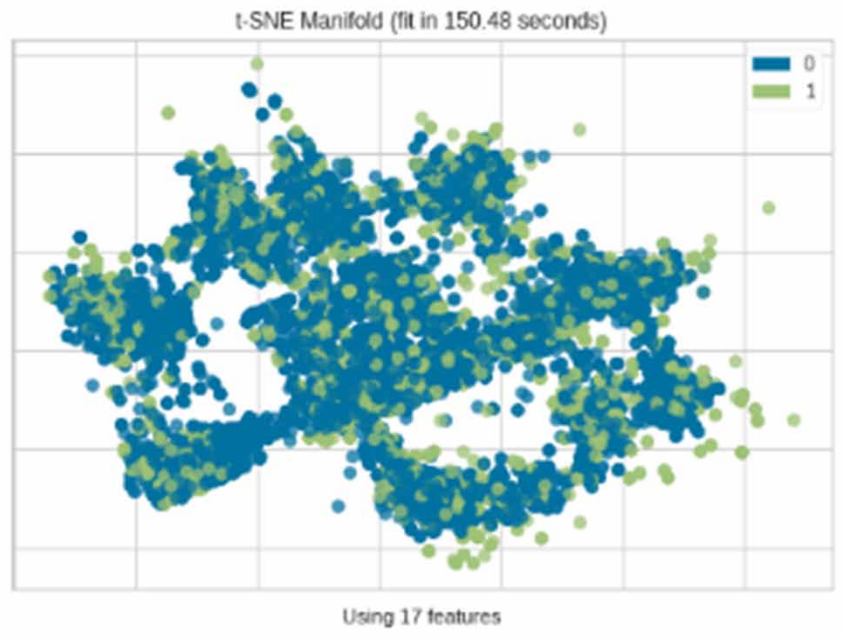


Figure 18(d) Applying DT (d) Calibration curve

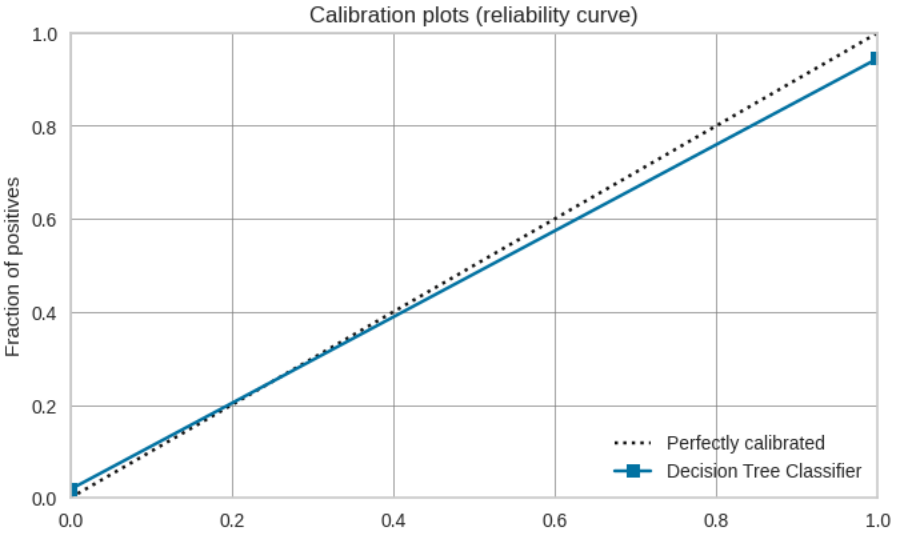


Figure 19(a). Applying LightGBM (a) ROC curves

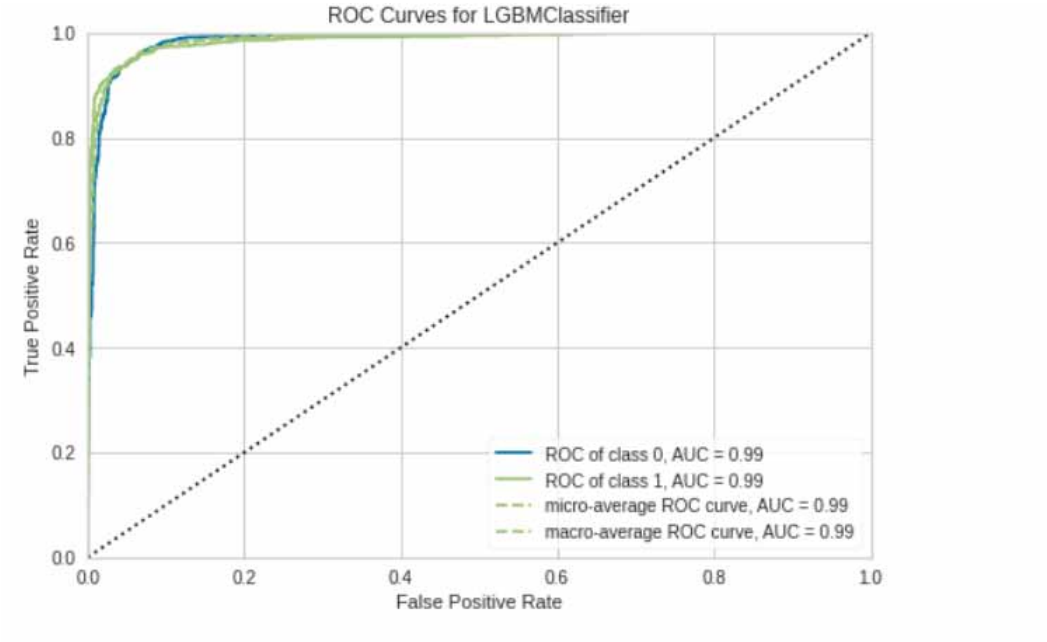


Figure 19(b). Applying LightGBM (b) Confusion matrix

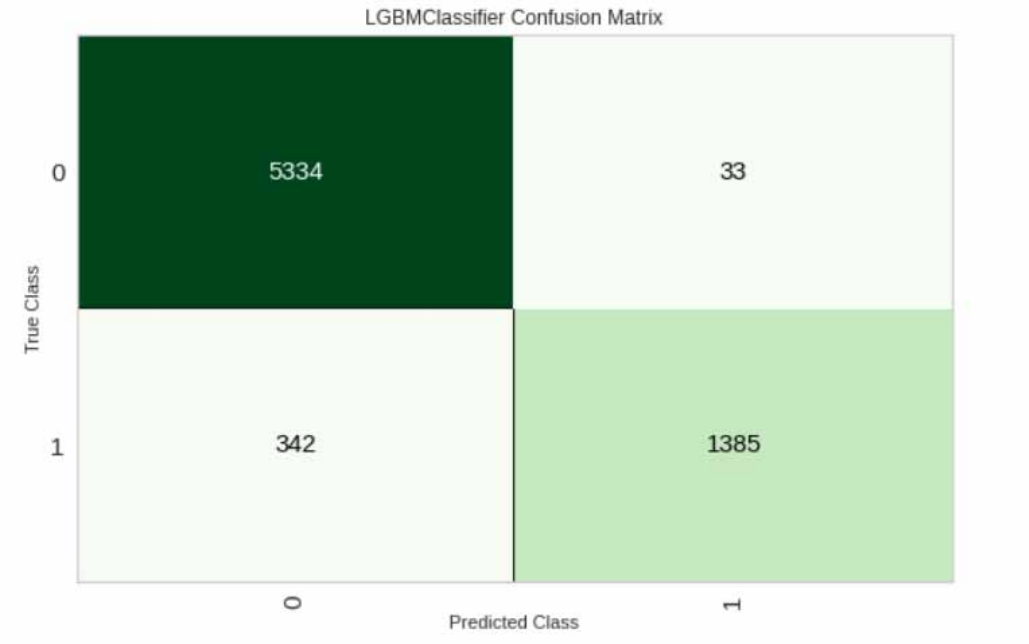


Figure 19(c). Applying LightGBM (c) t-sne manifold curves

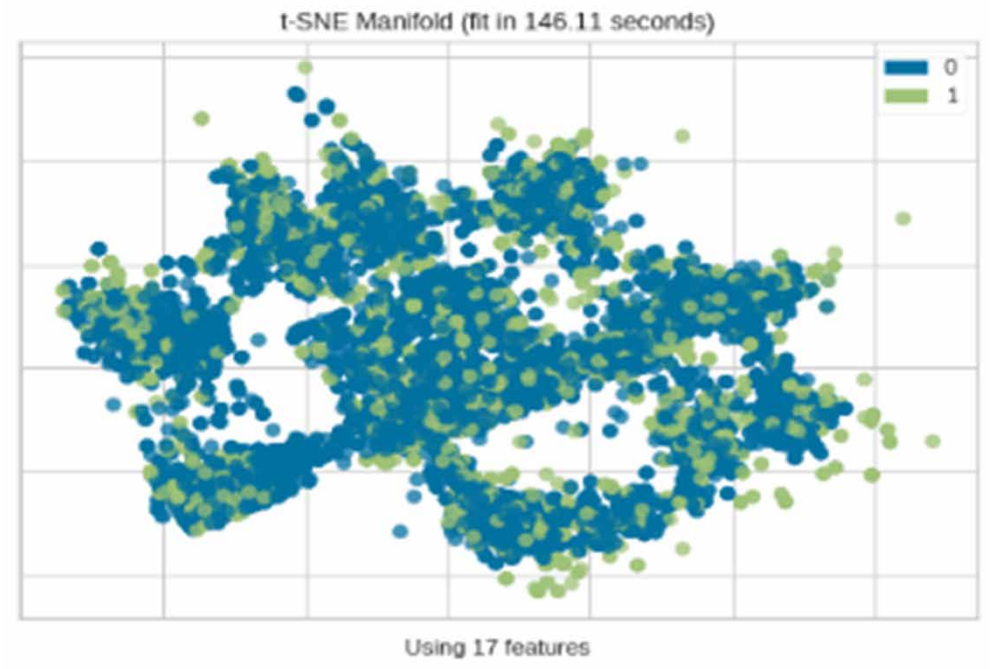


Figure 19(d). Applying LightGBM (d) Calibration curve

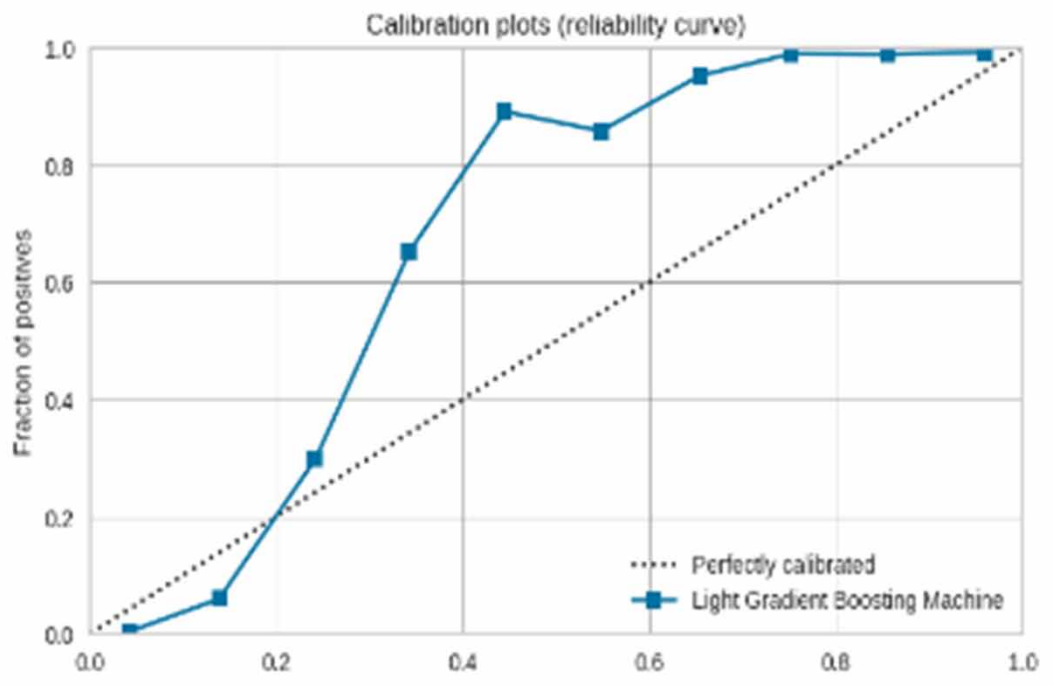


Figure 20(a). Applying KNN (a) ROC curves

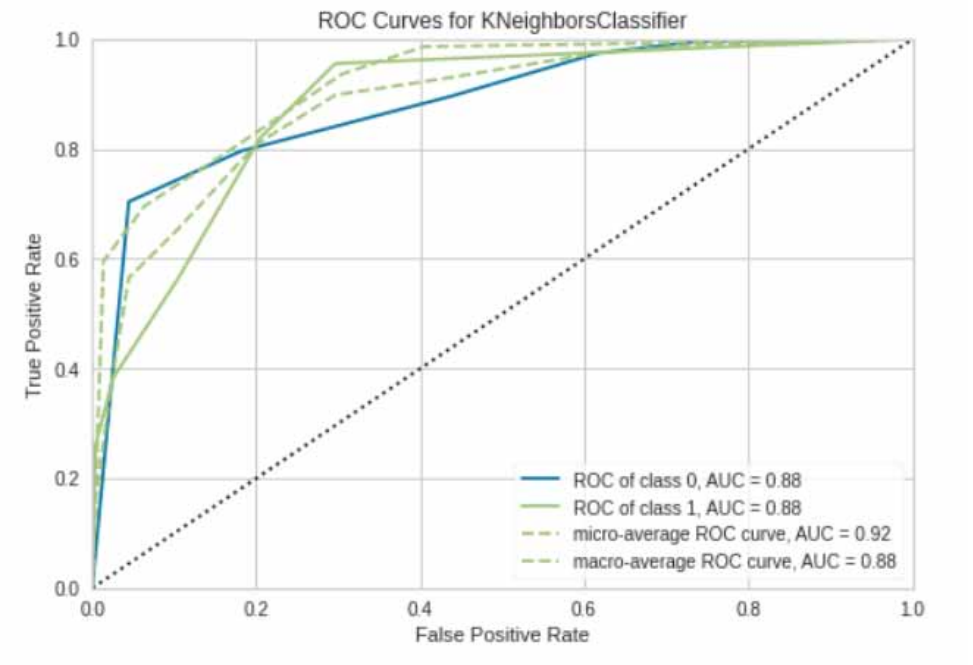


Figure 20(b). Applying KNN (b) Confusion matrix

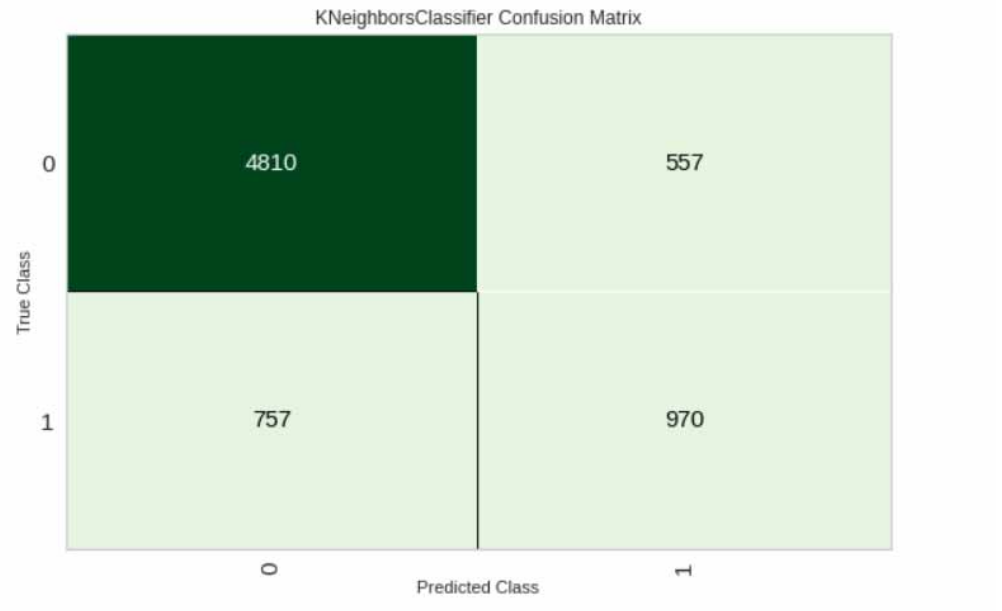


Figure 20(c). Applying KNN (c) t-sne manifold curves

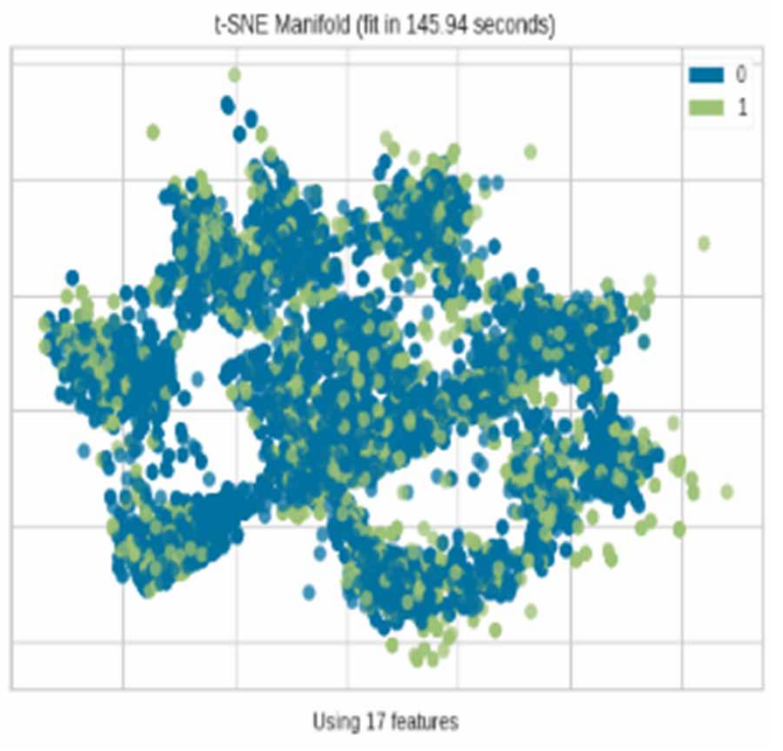


Figure 20(d). Applying KNN (d) Calibration curve

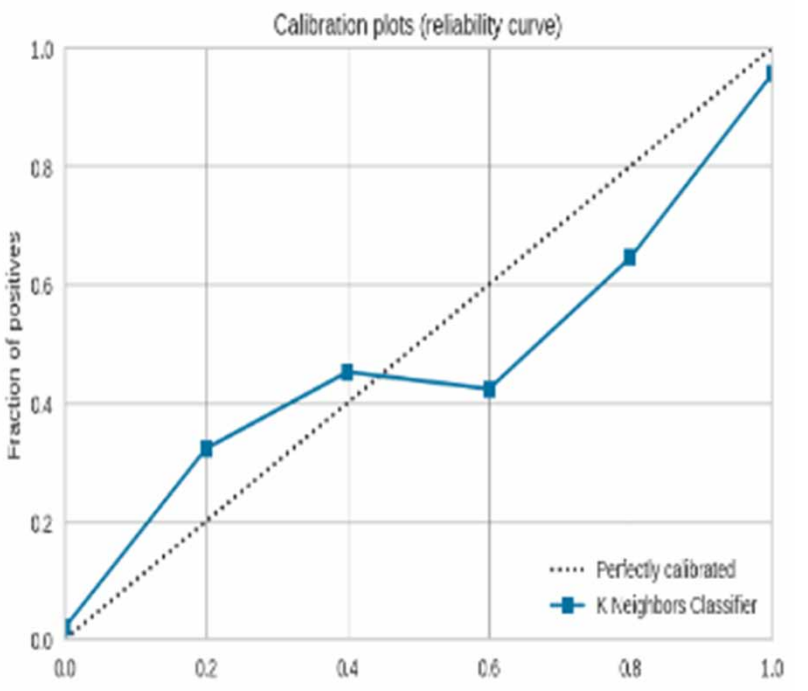


Figure 21(a). Applying LR (a) ROC curves

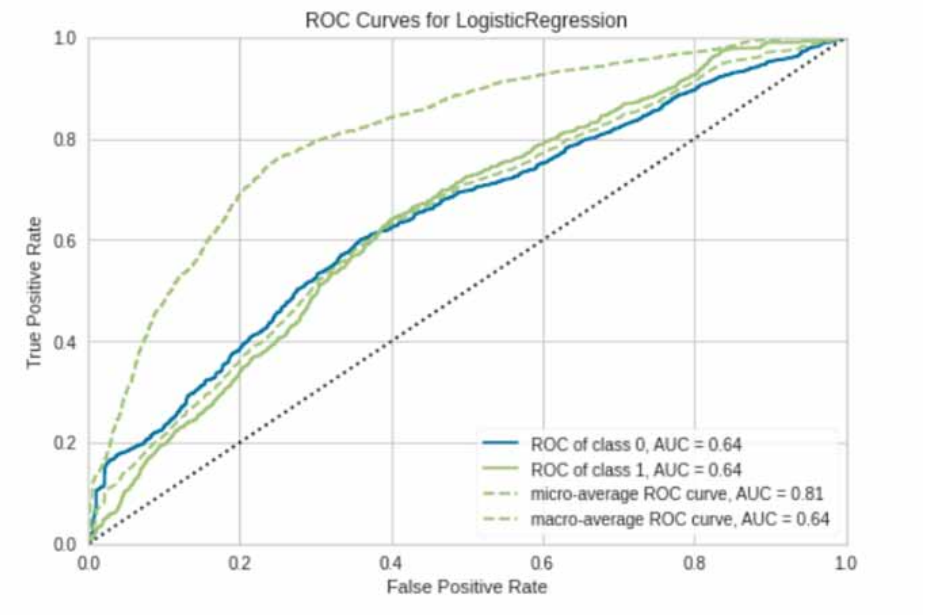


Figure 21(b). Applying LR (b) Confusion matrix

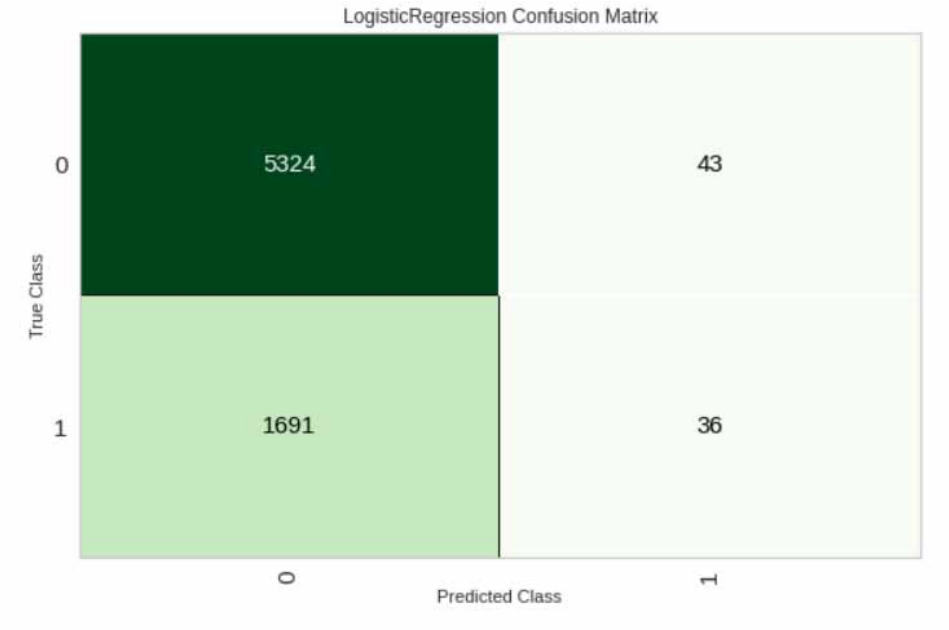


Figure 21(c). Applying LR (c) t-sne manifold curves

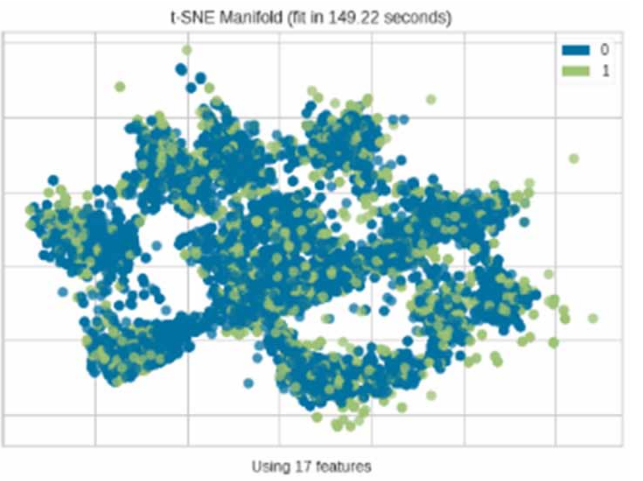
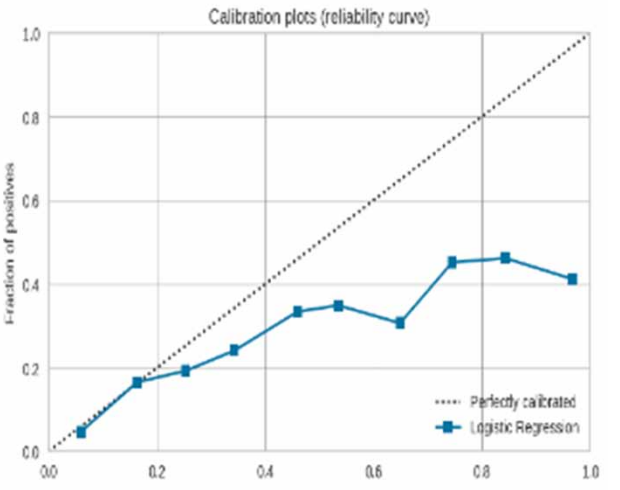


Figure 21(d). Applying LR (d) Calibration curve



Applying DT, Fig 18(a) well skilled ROC can be seen having 5272 TP, 95 FN, 99 FP, 1628 TN values. In Fig 18(c) t-sne is done with 17 features, differentiating the anomalous and non-anomalous classes. Fig 18(d) represents the perfectly calibrated curve. In contrast to RF, LightGBM shows a difference with more FN and FP values in Fig 19(b) and also the calibration plot is poorly depicted as seen in Fig 19(d). KNN and LR shows atrocious results compared to other models which can be seen in Fig 20(a) and Fig 21(a), also in Fig 20(b), Fig 21(c) shows CM for both the models have more errors.

V. CONCLUSION AND FUTURE WORK

With the enormous increase in data production, anomaly detection plays a prominent role in the finer analysis process. Industrial Internet of Things (IIoT) or Industrial IoT that at first chiefly alluded to a mechanical system whereby an enormous number of devices or machines are associated and synchronized using programming devices and third stage advancements in a machine-to-machine and Internet of Things, later an Industry 4.0 or Industrial Internet of Things. The data produced by multiple huge numbers of sensors are incredibly complicated, diverse, and massive in IIoT and is raw. These may contain anomalies which are needed to be identified for better data analysis. In this research, we have compared the existed algorithms of classification for detecting anomalies in IIOT data. The algorithms being compared here are Random Forest (RF), Logistic Regression (LR), Light Gradient Boosting Machine (LightGBM), Decision Trees (DT), K Nearest Neighbors (KNN).

We have done a comparative analysis of five existing classification algorithms to detect the anomalies. Four IIoT datasets were considered and the experiments were conducted using the PyCaret library. The results show that Random Forest (RF) gave best results of 99% and 98% accuracies for Demand vs Response and E-filtration painting maintenance datasets respectively. For the semiconductor manufacturing dataset, KNN has shown 94% accuracy along with RF 93.98% which is close to 94%. For the HRSS dataset, RF shows best results in depicting anomalies with an accuracy of 98.15%. Hence, the Random Forest algorithm shows impressive results for multidimensional IIoT datasets.

For future works, algorithms can be blended to make an ensembled model which targets to provide more accurate results on this type of large datasets.

FUNDING AGENCY

Publisher has waived the Open Access publishing fee.

REFERENCES

Hassanzadeh, A., Modi, S., & Mulchandani, S. (2015). Towards Effective Security Control Assignment in the Industrial Internet of Things. *IEEE 2nd World Forum on Internet of Things (WF-IoT).* doi:10.1109/WF-IoT.2015.7389155

Genge, B., Haller, P., & Enachescu, C. (2019). Anomaly Detection in Aging Industrial Internet of Things. *IEEE Access: Practical Innovations, Open Solutions*, *7*, 1–1. doi:10.1109/ACCESS.2019.2920699

Chahla, C., Snoussi, H., Merghem, L., & Esseghir, M. (2019). A novel approach for anomaly detection in power consumption data. *Proc. 8th Int. Conf.Pattern Recognit. Appl. Methods*. doi:10.5220/0007361704830490

Di Wu, Z. J., Xie, X., Wei, X., Yu, W., & Li, R. (2019). LSTM Learning with Bayesian and Gaussian Processing for Anomaly Detection in Industrial IoT. *IEEE Transactions on Industrial Informatics*. Advance online publication. doi:10.1109/TII.2019.2952917

Sisinni, E., Saifullah, A., Han, S., Jennehag, U., & Gidlund, M. (2018). Industrial Internet of Things: Challenges, Opportunities, and Directions. *IEEE Transactions on Industrial Informatics, 14*(11). https://towardsdatascience.com/supervised-machine-learning-technique-for-anomaly-detection-logistic-regression-97fc7a9cacd4

Razzak, Zafar, Imran, & Xu. (2020). Randomized nonlinear one-class support vector machines with bounded loss function to detect of outliers for large scale IoT data. *Future Generation Computer Systems, 112,* 715-723.

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H., & Zhao, W. (2017). A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications. *IEEE Internet of Things Journal, PP*, *4*(99), 1–1. doi:10.1109/JIOT.2017.2683200

Da Xu, L., He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics*, *10*(4), 2233–2243. doi:10.1109/TII.2014.2300753

Schreyer, M., Sattarov, T., Borth, D., Dengel, A., & Reimer, B. (2018). *Detection of anomalies in large scale accounting data using deep autoencoder networks*. arXiv:1709.05254.

Hasan, Islam, Zarif, & Hashem. (2019). Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches. *Internet of Things, 7.* www.elsevier.com/locate/iot

Nesa, Ghosh, & Banerjee. (2018). Non-parametric sequence-based learning approach for outlier detection in IoT. *Future Generation Computer Systems, 82,* 412-421.

Abdel Rahman, O., & Keikhosrokiani, P. (2020). Assembly Line Anomaly Detection and Root Cause Analysis Using Machine Learning. *IEEE Access, Digital Object Identifier*, *8*(October), 189661–189672. Advance online publication. doi:10.1109/ACCESS.2020.3029826

Zhu, Ji, Yu, Tan, Zhao, Li, & Xia. (2020). KNN-Based Approximate Outlier Detection Algorithm Over IoT Streaming Data. In *Special Section on Innovation and Application of Internet of Things and Emerging Technologies in Smart Sensing*. IEEE Access.

Cai, S., Li, L., Li, S., Sun, R., & Yuan, G. (2020). An efficient approach for outlier detection from uncertain data streams based on maximal frequent patterns. *Expert Systems with Applications, 160.* www.elsevier.com/locate/eswa

Cui, W., & Wang, H. (2017, November). A new anomaly detection system for school electricity consumption data. *Information (Basel)*, *8*(4), 151. doi:10.3390/info8040151

Liu, X., & Nielsen, P. S. (2016). *Regression-based online anomaly detection for smart grid data*. arXiv:1606.05781.

Xu, Liu, Yao, & Li. (2018). A Comparison of Outlier Detection Techniques for High-Dimensional Data. *International Journal of Computational Intelligence Systems, 11,* 652–662.

Yang, H., Liang, S., Ni, J., Li, H., & Shen, X. (2020). Secure and Efficient kNN Classification for Industrial Internet of Things. IEEE Internet of Things Journal, 7(11), 10945 – 10954.

Sivadi Balakrishna is currently working as Assistant Professor in Computer Science & Engineering, VIGNAN's Foundation for Science, Technology & Research (Deemed to be University), Guntur, AP, India. He has qualified UGC-NET in Dec-2018. He has published more than 15 research articles in international journals and contributed chapters to several books. He has also presented papers at several international conferences. His current research interests are Machine Learning, Internet of Things, and AI.