Exploiting Boosting Phenomenon for Classification Problems

A Thesis

submitted in partial fulfilment of the requirement for the

award of degree

of

MASTER OF TECHNOLOGY IN COMPUTER TECHNOLOGY

In the Faculty of Engineering and Technology

Jadavpur University

By

SOURABH GHOSH

Registration No.: 154182 of 2020-21

Class Roll No.: 002010504016

Examination Roll No.: M6TCT23026

Under the Guidance of

Prof. Susmita Ghosh

Department of Computer Science and Engineering

Jadavpur University, Kolkata-700032

India

2023

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

CERTIFICATE

This is to certify that the dissertation entitled "**Exploiting Boosting Phenomenon for Classification Problems"** been carried out by Sourabh **Ghosh** (University Registration No.- 154182 of 2020-21, Class Roll No. 002010504016) under my guidance and supervision and to be accepted in partial fulfilment of the requirement for the Degree of **Master of Technology in Computer Technology** in the Department of Computer Science and Engineering in the Faculty of Engineering and Technology, Jadavpur University, during the academic year 2022-23 and having Examination Roll No: M6TCT23026.

Prof. Dr. Susmita Ghosh (Thesis Supervisor)
Professor, Dept. of Computer Science & Engineering
Jadavpur University, Kolkata 700032

Prof. Nandini Mukhopadhyay
Head, Department of Computer Science and Engineering
Jadavpur University, Kolkata 700032

Prof. Saswati Mazumdar
Dean, Faculty of Engineering and Technology
Jadavpur University, Kolkata 700032

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Certificate of Approval

This is to certify that the thesis entitled "**Exploiting Boosting Phenomenon for Classification Problems**" is a bonafide record of work carried out by **Sourabh Ghosh** in partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Technology** in the Department of Computer Science and Engineering, Jadavpur University. It is understood that by this approval the undersigned do not necessarily endorse or approve any statement made, opinion expressed OR conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

………………………………………

Signature of Examiner

Date:

………………………………………

Signature of Supervisor

Date:

FACULTY OF ENGINEERING AND TECHNOLOGY

JADAVPUR UNIVERSITY

Declaration of Originality and Compliance of Academic Ethics

I hereby declare that this thesis entitled "**Exploiting Boosting Phenomenon for Classification Problems**" contains literature survey and original research work by the undersigned candidate, as part of his Degree of **Master of Technology in Computer Technology**. All information hasbeen obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Sourabh Ghosh

Registration No: 154182 of 2020-2021

Class Roll No.: 002010504016

Examination Roll No: M6TCT23026

Thesis Title: ***Exploiting Boosting Phenomenon for Classification Problems***

…..………………………………………..

Signature with Date

Acknowledgement

I would like to start by thanking my parents, my sister and my teachers for helping me deploy all the right resources and for shaping me into a better human being.

I would like to express my deepest gratitude to my advisor, Prof. Susmita Ghosh, Department of Computer Science and Engineering, Jadavpur University for her admirable guidance, care, patience, mental support and for providing me with an excellent atmosphere for doing research.

I am also thankful to university for giving me the proper laboratory facilities for carrying out my work. This thesis would not have been completed without the inspiration and support of a number of wonderful colleagues -- my thanks and appreciation to all of them for being part of this journey and making this thesis possible.

……………………………………………..

Sourabh Ghosh

Department of Computer Science & Engineering

Jadavpur University

Contents

List of Tables

CHAPTER 1

Introduction

The volume of data in the medical sector continues to grow significantly every day. Managing and extracting valuable insights from such vast datasets for effective decision-making poses a formidable challenge. To address this demand, the medical industry seeks specialized techniques capable of providing meaningful decisions from extensive databases. In today's medical landscape, classification models are transforming disease diagnosis and treatment. These AI-powered algorithms excel at analysing vast patient data, detecting subtle patterns, and accurately categorizing diseases [1]. By uncovering hidden patterns and valuable knowledge within large datasets, they offer an innovative approach to address real-world challenges. In the healthcare industry, where substantial data is collected from various clinical reports and patient manifestations, classification becomes an essential procedure for comprehensive data analysis and knowledge gathering from diverse perspectives. The major benefits of these models include enhanced diagnostic accuracy, early disease detection, personalized treatment plans, and support for medical research. By harnessing the power of classification models, doctors and researchers can revolutionize healthcare, providing more efficient and effective medical care to patients worldwide.

In such a scenario, boosting algorithms have emerged as a game-changer in medical diagnosis, revolutionizing disease identification and treatment. Leveraging the power of artificial intelligence, these algorithms play a pivotal role in analysing extensive patient data, unearthing hidden patterns, and precisely categorizing diseases. Their integration into medical practice brings forth numerous advantages, such as heightened diagnostic accuracy, early disease detection, personalized treatment plans, and accelerated medical research [17]. With the synergistic combination of boosting algorithms and medical expertise, healthcare professionals can provide more efficient and effective care, making a profound impact on patients' lives worldwide.

This study delves into the world of boosting algorithms, exploring their impact on system performance and responsiveness. We embark on a comparative analysis of five distinct boosting algorithms, each designed to optimize various aspects of the boosting process. Through an in-depth examination of these algorithms, we aim to provide valuable insights into their strengths and weaknesses, shedding light on their suitability for different computing environments and use cases.

CHAPTER 2

Boosting Algorithms

2.1 Boosting

Boosting in AI/ML is a meta-algorithm that combines multiple weak learning models to createa strong learning model that can make accurate predictions. These algorithms work by iteratively training a set of weak models, each of which performs better than random guessing but may still have high error rates. In each iteration, the algorithm assigns higher weights to the misclassified examples, which causes the subsequent models to focus more on these examples. The final prediction is then made by combining the weighted predictions of all the weak models. Boosting algorithms have been shown to improve the accuracy of machine learning models, reduce bias and variance, and increase robustness to outliers.

2.2 Why Boosting came?

Boosting in AI/ML has its roots in the work of computer scientists Robert Schapire and Yoav Freund, who first introduced the idea of boosting in 1990. Their work focused on developing a new learning algorithm that could improve the accuracy of weak learning algorithms. This algorithm, known as AdaBoost, quickly gained popularity and became a standard tool in machine learning. The Adaboost algorithm [1] is a type of boosting algorithm that iteratively trains weak models on the data, with each iteration adjusting the weights of the misclassified data points to prioritize their correct classification in the next iteration. Adaboost was shown to be effective in improving the accuracy of binary classification tasks, and it quickly gained popularity among AI/ML researchers.

Over the years, boosting has been refined and expanded upon, with new boosting algorithms being developed to address various limitations and challenges. For example, the Gradient Boosting Machine (GBM)[14] algorithm was introduced in 2001 by Jerome Friedman, and it is based on the idea of minimizing a cost function by iteratively adding weak models to the ensemble. GBM has been shown to be effective in a variety of tasks, including regression, classification, and ranking.

Another notable advancement in boosting is XGBoost [13], a high-performance implementationof GBM developed by Tianqi Chen and Carlos Guestrin in 2016. XGBoost is designed to be scalable and efficient, and it has been used to win several machine learning competitions on the Kaggle platform.

Boosting has also been applied to a variety of other AI/ML tasks beyond classification and regression, including object detection, image segmentation, and natural language processing. It has shown to be effective in these tasks as well, leading to significant advancements in these fields.

The advent of boosting algorithms marked a significant breakthrough in the world of Artificial Intelligence and Machine Learning (AI/ML), revolutionizing the landscape of classification tasks. Boosting algorithms emerged as a response to the limitations of traditional single-model approaches, aiming to enhance the predictive power and accuracy of machine learning models for complex datasets. Their popularity soared due to their exceptional performance in tackling a wide array of classification challenges, effectively becoming one of the cornerstones of modern AI/ML.

Boosting algorithms gained traction primarily due to their ability to address the inherent weaknesses of individual weak learners. Weak learners, while having a limited predictive capability, could still contribute valuable insights. The genius of boosting lay in its ensemble approach, wherein multiple weak learners were sequentially trained to focus on the misclassified instances of the previous models. Through this iterative process, boosting algorithms aimed to correct errors and improve overall classification accuracy.

The main reason behind the unprecedented success of boosting algorithms was their proficiency in creating strong learners from a pool of weak ones. The sequential nature of boosting, exemplified by algorithms like AdaBoost and Gradient Boosting, allowed each subsequent learner to emphasize the misclassified instances, effectively "boosting" the model's performance. This iterative nature ensured that the model learned from its mistakes, gradually honing in on complex decision boundaries that single models struggled to capture.

Furthermore, boosting algorithms exhibited impressive generalization capabilities. By preventing overfitting and reducing bias, boosting struck a delicate balance between model complexity and predictive accuracy. This not only improved the algorithm's ability to handle noisy data but also facilitated its adaptation to various types of classification problems, ranging from image recognition to natural language processing.

In essence, the success of boosting algorithms in AI/ML classification was attributed to their ensemble approach, adaptive learning, and ability to harness the strengths of multiple weak learners. By iteratively refining predictions and combining the insights of individual models, boosting algorithms pushed the boundaries of classification accuracy, opening new avenues for solving complex real-world problems. Their ability to amplify performance through collaboration transformed them into indispensable tools in the AI/ML practitioner's toolkit, forever changing the way we approach classification tasks.

So, it can be seen that, boosting has come a long way since its inception in 1990, with many advancements and refinements in the algorithms used, as well as new applications and use cases. Overall, the progression of boosting in AI/ML has been marked by a steady increase in the accuracy and efficiency of boosting algorithms. It continues to be a popular and effective technique in AI/ML, with researchers actively exploring new ways to apply and improve it.

2.3 Importance of Boosting in AI/ML

Boosting is an important technique in AI/ML due to its ability to improve the accuracy and robustness of machine learning models. Boosting algorithms work by iteratively combining multiple weak models to create a stronger model that can make more accurate predictions.

One of the main advantages of boosting is that it reduces bias and variance in machine learning models. By combining multiple weak models, boosting algorithms can effectively reduce the risk of overfitting and underfitting. This makes boosting particularly useful in situations where data is noisy or contains outliers.

Another important aspect of boosting is its versatility. Boosting algorithms can be applied to a wide range of problems in different domains, including computer vision, natural language processing, and predictive modeling. Furthermore, boosting algorithms can be used with various types of data, including structured, unstructured, and semi-structured data.

In addition, boosting is an important tool for handling imbalanced datasets. Imbalanced datasets occur when one class in a classification problem is more common than the other. Boosting algorithms can be used to handle imbalanced datasets by oversampling the minority class and undersampling the majority class, thereby improving the accuracy of the model.

Finally, boosting algorithms have been shown to outperform other machine learning algorithms, such as random forests and support vector machines, in many applications. This makes boosting an important technique in the field of AI/ML and a valuable tool for researchers and practitioners alike.

Overall, the importance of boosting in AI/ML lies in its ability to improve the accuracy and robustness of machine learning models, handle imbalanced datasets, and its versatility in various applications. Boosting is a powerful tool that can help researchers and practitioners to develop more accurate and efficient machine learning models, and it will continue to be an important technique in AI/ML in the years to come.

2.4 Different Boosting Algorithms

There are several boosting algorithms that have been invented over the years in the field of artificial intelligence and machine learning. The sequence in which they were invented is as follows:

2.4.1 Adaboost

The history behind AdaBoost, or Adaptive Boosting [1], [2], [3], dates back to the mid-1990s. The algorithm was first introduced in a paper by Yoav Freund and Robert Schapire in 1995. At the time, they were both researchers at AT&T Bell Laboratories, and they were working on improving the accuracy of classification algorithms.

The idea behind AdaBoost was inspired by an earlier boosting algorithm called Boosting by Committee, which was introduced in 1994 by David Haussler and others. Boosting by Committee involved training multiple classifiers on different subsets of the data and combining their predictions. However, the algorithm suffered from a high variance problem, where small changes in the data could result in significantly different outputs.

Freund and Schapire set out to address the high variance problem of Boosting by Committee by introducing a new boosting algorithm called AdaBoost. The name "AdaBoost" comes from the term "adaptive boosting", which refers to the algorithm's ability to adapt to the changing weights of the training examples. Their approach was to train a sequence of weak classifiers, each of which focuses on the examples that were misclassified by the previous classifiers. During each iteration, AdaBoost assigns higher weights to the misclassified examples, which forces the next weak classifier to pay more attention to these examples. This iterative process continues until the desired level of accuracy is achieved or until a predefined number of iterations is reached.

The initial results of AdaBoost were impressive, and the algorithm quickly gained popularity in the machine-learning community. In 2003, Freund and Schapire were awarded the prestigious Gödel Prize for their work on AdaBoost, which is considered one of the most important breakthroughs in machine learning and computational theory. Since then, AdaBoost has been used in various real-world applications, including computer vision, natural language processing, and bioinformatics.

AdaBoost Algorithm

2.4.1.1 Initialize weights for each data point as w(xi , yi) = 1 n , I = 1,….,n.

2.4.1.2 For iterations m= 1,,M

   2.4.1.2.1 Fit weak classifiers to the data set and select the one with the lowest weightedclassification error:

   $€m$= Ew[1y≠f(x)]

   2.4.1.2.2 Calculate the weight of the m classifier:

   θm= 1 2 ln(1−$\in m$ $\in m$)

2.4.1.3 Update the weight for each data point as:

   wm+1(xi ,yi) = (wm(xi ,yi)exp[$-\theta m$ $yifm$(xi)])/zm

   Where zm is a normalization factor that ensures the sum of all instance weights is equal to 1.

After M iteration we can get the final prediction by summing up the weighted prediction of each classifier.

One of the key advantages of AdaBoost is that it is relatively simple to implement and can be applied to a wide range of problems, including classification and regression tasks. It is also known for its ability to handle noisy data and outliers, and for its robustness to overfitting.

However, like any algorithm, AdaBoost has its limitations. It can struggle with large datasets or datasets with high dimensionality. Also, because AdaBoost relies on a sequence of weak classifiers, it can be sensitive to the quality of these individual classifiers. Nevertheless, AdaBoost remains an important algorithm in machine learning and is widely used in various real-world applications.

2.4.2 Gradient Boosting

1. Let’s assume X, and Y are the input and target having N samples. Our goal is to learn the function f(x) that maps the input features X to the target variables y. It is boosted trees i.e the sum of trees.

   The loss function is the difference between the actual and the predicted variables.

$$L(f) = \sum_{i-1}^{N} L(y_i, f(x_i))$$

2. We want to minimize the loss function L(f) with respect to f.

$$\hat{f}_0 = argmin_f \ L(f) = argmin_f \sum_{i-1}^{N} L(y_i, f(x_i))$$

   If our gradient boosting algorithm is in M stages then To improve the f_m algorithm can add some new estimator as h_m having 1 <m < M

$$\hat{y}_i = F_{m+1}(x_i) = F_m(x_i) + h_m(x_i)$$

3. Steepest Descent

   For M stage gradient boosting, The steepest Descent finds h_m = -ρ_m g_m where ρ_m is constant and known as step length and g_m is the gradient of loss function L(f)

$$g_{im} = - \frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}_{f(x_i) = f_{m-1}(x_i)}$$

4. Solution

   The gradient similarly for M trees:

$$(\rho_m, h_m(x)) = \arg \min_{\rho, h} \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + \rho h(x_i))$$

The current solution will be

$$f_m = f_{m-1} - \rho_m \ g_m$$

Boosting techniques amalgamate weak learners, which are marginally more proficient than random guessing, into a robust learner through an iterative process (Yoav Freund, 1999). Gradient boosting represents a variant of boosting algorithms applied to regression problems (Friedman, 2001). Given a training dataset $D=\{x_i, y_i\}_1^N$, the goal of gradient boosting is to find an approximation, $\hat{F}$ (x), of the function F^* (x), which maps instances x to their output values y, by minimizing the expected value of a given loss function, L(y, F(x)). Gradient boosting builds an additive approximation of F^* (x), as a weighted sum of functions

$$F_m(x) = F_{m-1}(x) + \rho_m \, h_m(x)$$

where ρ_m is the weight of the mth function, h_m (x). These functions are the models of the ensemble (e.g., decision trees). The approximation is constructed iteratively. First, a constant approximation of F^* (x) is obtained as

$$F_0(x) = \arg \min_a \sum_{i=1}^{N} L(y_i, a)$$

Subsequent models are expected to minimize

$$(\rho_m , h_m(x)) = \arg \min_{\rho,h} \sum_{i=1}^{N} L(y_i , F_{m-1}(x_i) + \rho h(x_i))$$

However, instead of solving the optimization problem directly, each h_m can be seen as a greedy step in a gradient descent optimization for F∗. For that, each model, h_m, is trained on a new dataset $D=\{x_i, r_{mi}\}_1^N$, where the pseudo-residuals, r_{mi}, are calculated by

$$r_{mi} = \frac{\partial L(y_i, F(x)}{\partial F(x)}_{F(x)=F_{m-1}(x)}$$

The value of ρ_m is subsequently computed by solving a line search optimization problem.

This algorithm might face issues related to overfitting if the iterative procedure isn't appropriately regularized (Friedman, 2001). In certain scenarios where loss functions (like quadratic loss) are used, if the model, hm, perfectly fits the pseudo-residuals, then in the subsequent iteration, the pseudo-residuals end up being zero, leading to premature termination of the process. To manage the step-by-step augmentation process of gradient boosting, multiple regularization hyperparameters are taken into account.

A natural approach to introduce regularization in gradient boosting involves applying shrinkage to diminish the impact of each gradient descent step, denoted as F_m (x) = $F_m - 1$ (x) + $\nu\rho_m$ h_m (x) where ν lies within the range of (0, 1.0]. Typically, the value of ν is set around 0.1. Furthermore, additional regularization can be accomplished by placing constraints on the complexity of the trained models. In the case of decision trees, it's feasible to restrict the depth of trees or the minimum instance requirement for node splitting.

2.4.3 XGBoost (Extreme Gradient Boosting)

XGBoost (Extreme Gradient Boosting) [13] is a popular algorithm for supervised machine learning tasks such as classification and regression. It was first proposed by Tianqi Chen and Carlos Guestrin in their 2016 paper "XGBoost: A Scalable Tree Boosting System."
The history of XGBoost can be traced back to gradient boosting, which is a technique for creating an ensemble of weak learners (e.g., decision trees) that work together to make accurate predictions. Gradient boosting works by iteratively adding new trees to the ensemble, with each new tree being trained to correct the errors of the previous trees.

XGBoost builds on this idea by introducing several novel techniques that improve the accuracy and efficiency of gradient boosting. XGBoost develops an additive extension of the objective function by minimising a loss function, similar to gradient boosting. Because XGBoost is only interested in decision trees as base classifiers, a loss function variation is employed to regulate the complexity of the trees.

$$Lxgb=\sum i=1NL(yi,F(xi))+\sum m=1M\Omega(hm)$$

$$\Omega(h)=\gamma T+12\lambda\|w\|2$$

where T is the number of leaves of the tree and w are the output scores of the leaves. This loss function may be implemented into decision trees' split criterion, resulting in a pre-pruning technique. Higher values of produce simpler trees. The value of determines the smallest loss reduction gain required to separate an internal node. Shrinkage is an extra regularisation hyper-parameter in XGBoost that decreases the step size in the additive expansion. Finally, the complexity of the trees can be restricted utilising different tactics such as tree depth, etc. A side advantage of reducing tree complexity is that models are trained faster and use less storage space. Randomization techniques are also used in XGBoost to prevent overfitting and speed up training. XGBoost includes random subsamples for training individual trees as well as column subsampling at the tree and tree node levels. Furthermore, by constructing a function that outputs the gradient and hessian (second order gradient) and sending it through the "objective" hyper-parameter, XGBoost may be extended to any user-defined loss function.

XGBoost suggests a sparsity-aware method for determining the optimal split. The existence of multiple zero-valued entries and/or missing values might lead an attribute to be sparse. For split candidates, XGBoost automatically eliminates these items from the gain computation. XGBoost trees also learn the default child node from which instances with missing or null values are branched. Other intriguing XGBoost features include monotonic and feature interaction limitations. When domain-specific information is available, these characteristics can be especially beneficial. Monotonic constraints force XGBoost's regression output to be monotonic (growing or decreasing) in relation to any set of supplied input characteristics. The input characteristics that can be mixed in the routes from the root to any leaf node are limited by feature interaction limitations. Both limitations are enforced by restricting the number of candidate splits examined at each node.

Furthermore, XGBoost includes a number of strategies for increasing the training speed of decision trees that are unrelated to ensemble accuracy. XGBoost, in particular, focuses on minimising the computational complexity for determining the optimal split, which is the most time-consuming element of decision tree building algorithms. Typically, split discovery algorithms list all potential candidate splits and choose the one with the biggest gain. To discover the optimal split for each node, a linear scan over each sorted attribute is required. To avoid having to sort the data in each node, XGBoost employs a compressed column-based structure in which the data is kept pre-sorted. As a result, each attribute only has to be sorted once. This column-based storage structure enables finding the appropriate split for each investigated characteristic in simultaneously. Furthermore, rather than scanning all potential candidate splits, XGBoost employs an approach based on data percentiles in which just a sample of candidate splits is examined and their gain is determined using aggregated statistics. This concept is similar to the node-level data subsampling that is already present in CART trees.

XGBoost has become a popular algorithm in the machine learning community and has been used to win numerous Kaggle competitions, as well as being widely adopted in industry applications.

2.4.4 LightGBM

LightGBM (Ke et al. 2017) [25], stands out as a comprehensive library that not only implements gradient boosting but also proposes several innovative variations. The core objective of this library's gradient boosting implementation is to ensure computational efficiency. It achieves this by precomputing feature histograms, akin to the approach taken by XGBoost. Within LightGBM, a rich set of learning hyperparameters is available, enabling the model's adaptability across diverse scenarios.

What sets LightGBM apart is its dual compatibility with both GPU and CPU, making it versatile in terms of hardware utilization. The library caters to basic gradient boosting needs while offering a range of randomization techniques such as column randomization and bootstrap subsampling.

LightGBM's implementation introduces novel features (Ke et al. 2017). These features primarily encompass Gradient-based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB). GOSS serves as a subsampling technique applied to construct training sets for the foundational trees within the ensemble. Drawing inspiration from AdaBoost, GOSS emphasizes instances with greater classification uncertainty, determined by higher gradients. When GOSS is activated, the training sets for base learners comprise a fraction of instances exhibiting the highest gradients (a), alongside a random sample fraction (b) drawn from instances with the lowest gradients. To counterbalance the shift in the original distribution, instances in the low gradient category receive a weight boost of (1 - a) / b during information gain computation.

Incorporating the Exclusive Feature Bundling (EFB) technique, LightGBM consolidates sparse features into a singular feature. This consolidation occurs without any loss of information, provided these features lack simultaneous nonzero values. Both GOSS and EFB contribute to expedited training speeds. It's worth noting that EFB functions as a preprocessing technique for features, which is why it won't be delved into within this study, as it's applicable beforehand to any dataset.

2.4.5 CatBoost (Categorical Boosting)

CatBoost [30] is an open-source gradient boosting framework specifically designed to handle categorical features. It was developed by Yandex, a Russian multinational corporation specializing in internet-related services and products, and was first released in 2017.

The idea for CatBoost originated from a problem that Yandex faced in their advertising business. They needed to develop a model to predict which ads would be clicked on by users, but their data contained many categorical features, such as the user's location or the type of device they were using. Traditional gradient boosting algorithms had difficulty handling these types of features, as they required converting categorical variables into numerical ones, which could result in information loss and decreased performance.

To overcome this problem, Yandex researchers developed a novel algorithm that could effectively handle categorical features in the gradient boosting framework. The algorithm uses several techniques, including a novel method for dealing with missing values in categorical features and a novel metric for evaluating the quality of splits. They also introduced a new way of handling categorical features, called ordered boosting, which uses an ordinal numbering scheme to represent the levels of each categorical variable.

The result of this research was the development of CatBoost, which quickly gained popularity among data scientists due to its ability to handle categorical features effectively and outperform other state-of-the-art algorithms on many datasets.

CatBoost is a gradient boosting library that seeks to reduce the shift in prediction that occurs during training. This distribution shift is the difference between the training instance F(xi)|xi and the test instance F(x)|x. This transition occurs because gradient boosting uses the same instances during training to estimate both gradients and the models that minimise those gradients. The solution proposed by CatBoost is to estimate gradients using a series of base models that do not include the instance in their training set. To accomplish this, CatBoost first introduces a random permutation to the training instances. CatBoost's concept (not its implementation) is to construct i=1,...,N base models for every M boosting iteration. The ith model of the mth iteration is trained on the initial i instances of the permutation and then used to estimate the gradient of the i+1 instance for the (m+1)th boosting iteration. This procedure is repeated using s distinct random permutations in order to be independent of the initial random permutation. Despite this, CatBoost is implemented in such a way that a single model is constructed per iteration that manages all possible permutations and models. The foundational models consist of symmetric trees (or decision tables). All leaf nodes are extended level-wise on these trees using the same split condition. CatBoost's handling of categorical features is another essential aspect of the system. CatBoost replaces categorical features with numeric features that quantify the expected objective value for each category. This numeric feature should ideally be computed using a distinct dataset to avoid overfitting to the training data. Nevertheless, this is typically not feasible. CatBoost's proposed method for calculating this new feature is analogous to the one used in model development. In other words, for a given random permutation of instances, the information of instance i is used to compute the feature value of instance i. The obtained feature value for each instance is then averaged after a series of permutations.

CHAPTER 3

Literature Survey

Boosting algorithms represent a pivotal facet of modern machine learning, standing as robust tools for enhancing the predictive performance of weak learners. Over the years, they have gained substantial traction across a plethora of domains due to their remarkable ability to adapt, learn, and amalgamate the strengths of multiple weak models into a potent ensemble. This literature survey delves into the expansive landscape of boosting algorithms, dissecting their evolution, underlying principles, and diverse applications. From the seminal AdaBoost to the more recent innovations like Gradient Boosting, XGBoost, LightGBM, and Catboost this comprehensive exploration seeks to provide a holistic understanding of the conceptual foundations, algorithmic intricacies, and practical utility of these techniques in the pursuit of predictive modeling excellence.

Following their initial independent contributions to boosting algorithms, Freund and Schapire introduced the adaptive boosting algorithm, widely known as AdaBoost, as documented in references [1], [2] and [3]. AdaBoost's fundamental concept revolves around the utilization of weighted instances within the same training dataset rather than opting for random subsampling. This approach operates through an iterative process, sequentially training a series of weak learners on a weighted version of the dataset. Each subsequent learner focuses on the samples that were misclassified by its predecessors. This iterative ensemble construction process leads to a robust and accurate final model.

AdaBoost's efficacy extends to text classification tasks. For instance, [4] utilized AdaBoost to enhance the performance of text categorization systems, demonstrating its aptitude in handling complex language data. In the realm of medical diagnosis, AdaBoost has found application in improving the accuracy of disease prediction models. Thongkam et al. (2008) conducted a comprehensive study on its use in medical diagnostics, highlighting its potential for aiding healthcare professionals in accurate disease prognosis [5]. Viola and Jones (2001) presented groundbreaking work in face detection, harnessing AdaBoost to select and combine Haar-like features for rapid and accurate face detection. This pivotal research has paved the way for facial recognition systems and applications [6].

Beyond the classic AdaBoost algorithm, several variations and improvements have emerged. The Real AdaBoost algorithm, presented by Friedman et al. (2000), introduced more robust weighting of instances, making it applicable in diverse scenarios [7]. Tanha et al. (2012) conducted a thorough exploration of AdaBoost's applicability in multi-class classification scenarios. This research expanded the algorithm's utility, allowing it to handle complex classification tasks involving multiple classes, thereby broadening its practicality [8].

Understanding and harnessing ensemble diversity is a critical aspect of boosting. Researchers like Zhu et al. (2009) have delved into methods for enhancing diversity among the base learners in AdaBoost. By promoting diverse learning approaches, the algorithm can further improve its generalization performance [9]. AdaBoost has proven valuable in object recognition tasks too. For instance, Lienhart et al. (2002) applied AdaBoost to real-time face detection and recognition, paving the way for applications in security and human-computer interaction [10].

AdaBoost has been combined with other machine learning techniques to create hybrid models. Researchers like Galar et al. (2012) explored hybrid models combining AdaBoost with support vector machines for improved classification accuracy [11]. AdaBoost.RT, introduced by Sapienza and Feelders (2005), extended AdaBoost to regression tasks. This adaptation has been applied in financial forecasting and other regression problems [12].

Gradient Boosting, a powerful ensemble learning technique, has gained immense popularity in the field of machine learning due to its exceptional predictive capabilities and versatility. It operates by iteratively training a series of weak learners, typically decision trees, to correct the errors of their predecessors, thus constructing a robust and accurate ensemble model. While the concept of boosting was introduced by Freund and Schapire in the late 1990s, Gradient Boosting, as we know it today, has witnessed significant advancements and adaptations. This literature survey embarks on a journey to explore the extensive body of work that has harnessed Gradient Boosting across a myriad of domains, showcasing its evolution, unique variants, and profound contributions to the machine learning landscape.

Gradient Boosting has found extensive application in predictive modeling. Researchers like Chen and Guestrin (2016) showcased its prowess in regression tasks, particularly in areas like house price prediction and financial forecasting [13]. Gradient Boosting excels in classification problems as well. For example, Hastie et al. (2009) introduced the Gradient Boosting Machine (GBM), which has become a standard choice for tasks such as spam detection and image classification [14].

In the realm of drug discovery, Gradient Boosting has been instrumental. Researchers like Chen et al. (2012) employed it for QSAR modeling, aiding in the development of new pharmaceutical compounds [15]. It has also played a pivotal role in disease diagnosis. Karabayir et al. (2020) utilized Gradient Boosting to diagnose medical conditions from patient data, facilitating early intervention and treatment [16]. In semantic segmentation of images also, Gradient Boosting has shown promise. Bakas et al. (2016) applied Gradient Boosting to improve the accuracy of glioma segmentation in medical images, making it a valuable tool for diagnosing and monitoring brain tumors. This research likely contributes to advancements in medical image analysis and the understanding of glioma growth and characteristics [17].

Efforts have been made to make Gradient Boosting models more explainable. Lundberg et al. (2018) proposed SHAP (SHapley Additive exPlanations), a framework for interpreting machine learning models, including Gradient Boosting, providing valuable insights into model predictions [18]. This approach can be applied to a wide range of applications, from healthcare and finance to natural language processing and computer vision, ultimately aiding in the transparency and trustworthiness of machine learning models.

In the financial sector, Gradient Boosting has been a game-changer in fraud detection. Gupta et al. (2019) employed it to detect fraudulent transactions in real-time, safeguarding financial institutions and their customers [19].

XGBoost's remarkable success can be attributed to its versatility and effectiveness in handling both regression and classification tasks. Its ability to handle missing data, feature selection, and regularization techniques, coupled with its scalability and speed, has made it a go-to algorithm for solving real-world problems, making it a popular choice among data scientists and machine learning practitioners.

In [13], Chen and Guestrin introduced XGBoost, providing a detailed explanation of the algorithm's inner workings and its superiority in terms of both speed and performance compared to other gradient boosting frameworks. This seminal work laid the foundation for subsequent research and applications of XGBoost. Ogunleye et al. demonstrated the effectiveness of XGBoost in the medical field by applying it to chronic kidney disease diagnosis [20]. Their study showcased how XGBoost could enhance accuracy in medical diagnoses, leading to potential improvements in patient care. Yang et al [21] investigated the use of XGBoost for anomaly detection. Their work highlighted the algorithm's capability to detect abnormalities in complex industrial systems, enhancing safety and efficiency. Zhang et al [22] explored the application of XGBoost in detecting fraudulent financial transactions. Their study demonstrated how XGBoost's ability to handle imbalanced datasets and capture subtle patterns made it a valuable tool for financial institutions to combat fraud.

In [23], Wu et al. discussed the application of XGBoost in drug discovery, showcasing its potential to expedite the identification of promising drug candidates and reduce research and development timelines. Su et al. utilized XGBoost for genomic analysis, specifically in identifying genetic markers associated with skin diseases [24]. This research offers insights into the potential applications of XGBoost in personalized medicine.

LightGBM (Light Gradient Boosting Machine) has emerged as a prominent and efficient gradient boosting framework, known for its speed and superior performance. This algorithm, developed by Microsoft, has gained popularity for its ability to handle large datasets and complex problems while maintaining impressive computational efficiency.

Ket et al in their seminal paper [25] introduces LightGBM, outlining its key features and demonstrating its efficiency in training and prediction. It presents a novel histogram-based learning method and benchmarks LightGBM against other gradient boosting frameworks, showcasing its superior speed and accuracy. In [26], Guo et al explored the application of LightGBM in solving multi-objective optimization problems, expanding its use beyond traditional regression and classification tasks. It introduces techniques for adapting LightGBM to tackle optimization challenges effectively.

Kopitar et al [27] applies LightGBM in the healthcare domain, focusing on risk prediction for diabetes. It also emphasizes the importance of interpretable machine learning models, showcasing LightGBM's ability to provide insights into predictive features.

Csizmadia et al [28] applied LightGBM with a multi-objective optimization approach to the field of human activity recognition. It showcases how LightGBM can be adapted to address complex, multi-objective problems. Yang et al in [29] explores the use of LightGBM for energy consumption forecasting. It highlights how LightGBM's speed and accuracy can help improve energy management and efficiency in this context.

CatBoost, short for Categorical Boosting, is a gradient boosting algorithm specifically designed to handle categorical features efficiently. Since its introduction, CatBoost has gained substantial attention from researchers and practitioners alike due to its superior performance in a wide range of applications.

The CatBoost algorithm was introduced by D. Prokhorenkova et al. in 2017 [30] . This seminal work laid the foundation for CatBoost, emphasizing its ability to handle categorical features naturally and the importance of its ordered boosting approach. The medical field recognized the potential of CatBoost. In 2021, Kumar et al. utilized CatBoost to build predictive models for diabetes prediction at early stages, illustrating its efficacy in predicting patient outcomes and disease diagnosis [31].

Natural Language Processing (NLP) applications have also seen the integration of CatBoost. In [32], Avelino et al. explored its use in NLP tasks such as doing sentiment analysis and text classification, detecting misinformation, showcasing its competitiveness with traditional NLP algorithms. In [33] and [34], CatBoost extended its reach into computer vision. Samat et al. introduced an approach using CatBoost for image classification tasks, indicating its versatility beyond tabular data.

In [35], Bo et al performed real-time prediction of the rock mass classification using the big data in the Songhua River Diversion Project, thus showcasing how promising Catboost algorithm is in the field of geological science.

These recent developments and applications highlight CatBoost's versatility and efficacy in solving real-world problems across a wide range of domains. Its adaptability, efficient handling of categorical features, and competitive performance make it a valuable asset for researchers and practitioners in the field of machine learning. As the algorithm matures and more insights are gained, CatBoost is likely to remain a prominent tool in addressing complex data-driven challenges.

CHAPTER 4

Results and Analysis

This study introduces boosting algorithms: Adaboost, Gradient Boosting, LightGBM, XGBoost, and CatBoost, which are utilized for the classification of datasets, basically belonging to the medical field. Additionally, the study compares the performance of these methods against each other. The research incorporates a total of 7 distinct datasets from different domains for this evaluation. To implement the proposed methodology in evaluating the classification models, training datasets were derived from existing dataset inventories. The target classes in each dataset were assigned labels such as "0", "1", and so on, corresponding to the classes in their respective datasets. The datasets were divided into training and testing sets accordingly for this purpose. The training set was used to train the proposed CatBoosting algorithm, as well as other boosting models like Adaboost and XGBoost.

The testing set played a crucial role in the comparative evaluation of the prediction models after the completion of the modeling stage. By employing this set, the performance of the models was meticulously assessed using multiple well-established performance metrics, namely Precision, Recall, Support, and F1 score, also referred to as F-score or F-measure. These metrics were selected for their ability to provide comprehensive insights into the models' predictive capabilities and effectiveness in handling the classification tasks on the diverse datasets from various fields. By utilizing these formal and widely recognized performance measures, the study sought to ensure a rigorous and objective assessment of the boosting algorithms, namely Adaboost, Gradient Boosting, LightGBM, XGBoost, and CatBoost, and their respective performance in classifying the datasets under investigation.

4.1 Datasets

A total of around 7 datasets were used for performing this analysis all of which were collected from UCI machine learning repository and also Kaggle, all of which are freely available for use. These datasets originate from various application domains and have varying numbers of attributes, classes, and instances.

i. **Breast Cancer Wisconsin Dataset**
   The features in this dataset are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass. They describe characteristics of the cell nuclei present in the image.[36]

This database is available through the UW CS ftp server [37].

Also, can be found on UCI Machine Learning Repository [38].

ii. **Heart Attack Analysis and Prediction Dataset**

This dataset on Kaggle is a comprehensive collection of health-related data aimed at exploring and predicting factors associated with heart attacks. The dataset encompasses various attributes that provide insights into an individual's health and lifestyle, allowing researchers and data scientists to analyze patterns and correlations related to heart health.

The dataset includes essential features such as age, gender, blood pressure, cholesterol levels, and heart rate, among others. These features play a significant role in understanding heart attack risk factors and potential predictive patterns. Additionally, the dataset includes the "output" column, which serves as the target variable and indicates whether a person has experienced a heart attack (1) or not (0).

This database is available on Kaggle [39].

iii. **Lung Cancer**

This dataset is a comprehensive collection of health-related information, encompassing 284 instances, each representing an individual patient. The dataset consists of 16 attributes that provide valuable insights into the patients' characteristics and potential risk factors associated with lung cancer.

The attributes include demographic information such as gender and age, lifestyle factors like smoking habits and alcohol consumption, and the presence of various symptoms like yellow fingers, anxiety, wheezing, chronic diseases, fatigue, allergies, coughing, shortness of breath, swallowing difficulty, and chest pain.

The primary objective of this dataset is to explore and analyze the potential relationships between these attributes and the occurrence of lung cancer. The target variable "Lung Cancer" indicates whether the patient has been diagnosed with lung cancer (YES) or not (NO).

This database is available on Kaggle [40].

iv. Pima Indiana Diabetes Dataset

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases.

The Pima Indian Diabetes dataset is a well-known and widely used collection of health-related data aimed at understanding diabetes occurrence among the Pima Indian population. The dataset contains information from 768 individuals, representing female Pima Indians aged 21 and above. The dataset encompasses several attributes, including age, number of pregnancies, glucose level, blood pressure, skinfold thickness, insulin level, body mass index (BMI), diabetes pedigree function, and an outcome variable indicating whether the individual developed diabetes within five years (1) or not (0).

This dataset is available on the UCI repository [41].

Also, it is available on Kaggle [42].

v. Dermatology

In dermatology, distinguishing between erythemato-squamous diseases poses a real challenge due to their shared clinical features of erythema and scaling, with minor differences. These diseases include psoriasis, seborrheic dermatitis, lichen planus, pityriasis rosea, chronic dermatitis, and pityriasis rubra pilaris. Accurate diagnosis often requires a biopsy, but unfortunately, these diseases also exhibit similar histopathological characteristics. Moreover, the early stages of a disease may resemble another, making differential diagnosis even more complex.

In the dataset constructed for this domain, the family history feature has the value 1 if any of these diseases has been observed in the family, and 0 otherwise. The age feature simply represents the age of the patient. Every other feature (clinical and histopathological) was given a degree in the range of 0 to 3. Here, 0 indicates that the feature was not present, 3 indicates the largest amount possible, and 1, 2 indicate the relative intermediate values.

This database can be found on the OpenML website [43].

It can be found on Kaggle [44].

vi. **Indian Liver Patient Records**

The prevalence of liver disease has been steadily rising due to factors such as excessive alcohol consumption, exposure to harmful gases, consumption of contaminated food, pickles, and drugs. This dataset was employed to assess prediction algorithms, aiming to alleviate the workload of medical professionals.

The Indian Liver Patient Dataset (ILPD) is a public dataset available on the UCI Machine Learning Repository. It is designed to aid in the study and analysis of liver disease prevalence and factors affecting it in the Indian population. The dataset includes a total of 583 instances (rows) with 10 attributes (columns), making it suitable for classification and regression tasks.

The dataset provides a binary class label, indicating whether the patient is a liver patient (1) or not (2).

This dataset is freely available on the UCI Repository [45].

Also, it can be found in Kaggle [46].

vii. **Parkinson Disease Detection**

Parkinson’s Disease (PD) is a degenerative neurological disorder caused by reduced dopamine levels in the brain, leading to movement issues like tremors and stiffness. Speech is also affected, with difficulties in articulation, volume, and pitch. Diagnosing PD traditionally relies on clinical evaluation and observation of motor skills, making early detection challenging due to the lack of definitive lab tests. An efficient screening process, like analyzing voice recordings with machine learning algorithms, could offer a non-invasive and effective way to diagnose PD before a clinic visit, facilitating early intervention and improved patient care.

This dataset consists of biomedical voice measurements from 31 individuals, 23 of whom have Parkinson's disease (PD). Each row represents one of the 195 voice recordings, identified by the "name" column. The table's columns represent specific voice measures. The primary objective of this data is to differentiate healthy individuals from those with PD. This distinction is indicated by the "status" column, where 0 denotes healthy individuals and 1 represents those with PD. The dataset consists of biomedical voice measurements from 31 individuals, 23 of whom have Parkinson's disease (PD). Each row represents one of the 195 voice recordings, identified by the "name" column. The table's columns represent specific voice measures. The primary objective of this data is to differentiate healthy individuals from those with PD. This distinction is indicated by the "status" column, where 0 denotes healthy individuals and 1 represents those with PD.

This dataset is available on the UCI Repository [47].

Also, it can be found on Kaggle [48].

4.2 Preprocessing

Data preprocessing is a crucial step in the implementation of machine learning algorithms as it plays a vital role in ensuring the accuracy, reliability, and efficiency of the model. By preparing and transforming the raw data into a suitable format, data preprocessing enables the algorithm to effectively learn from the information present in the dataset.

During the work:

1) We have dropped one column which only indicates the index value sequentially for all the records.

2) Then we have analysed the unique classes for each of the data. Those records who have missing values are eliminated.

3) Outliers have also been looked after.

In our work, for better performance, feature selection has been done in some of the datasets having large number of attributes. In Wisconsin Breast Cancer dataset, Information Gain strategy has been used to identify the most informative ones. Features with higher Information Gain contribute more to reducing the uncertainty in the target variable and are considered more relevant for the learning task. In dermatology dataset, the best features were selected from entire set of attributes by taking into consideration the correlation between them. Herein features that have a strong relationship with the target are retained, while discarding features that exhibit weak or no correlation.

4.3 Performance Metrics

a) Precision

Precision is a fundamental performance metric extensively used in the evaluation of classification models. It measures the accuracy of positive predictions made by the model, providing valuable insights into the model's ability to correctly identify true positives from the predicted positive instances. In essence, precision quantifies the proportion of true positive predictions relative to the total predicted positive instances.

Mathematically, precision is calculated as the ratio of true positives to the sum of true positives and false positives. A high precision value signifies that the model has a low rate of false positives, indicating a more reliable prediction of positive instances.

$$\text{Precision} = \frac{\text{True Positive(TP)}}{\text{True Positive(TP)} + \text{False Positive(FP)}}$$

Conversely, a low precision value implies a higher rate of false positives, highlighting the model's tendency to misclassify negative instances as positive.

Precision is particularly important in scenarios where false positives carry significant consequences or incur high costs. For instance, in medical diagnoses, a high precision model is crucial to avoid misdiagnosing healthy individuals as patients, as the consequences of such errors can be severe.

As a standalone metric, precision provides valuable information about the model's positive predictive power. However, it is often considered in conjunction with other metrics like recall, F1 score, and accuracy to gain a more comprehensive understanding of the model's overall performance. By carefully interpreting precision values alongside other performance measures, researchers and practitioners can make well-informed decisions about the efficacy and suitability of classification models for specific applications.

b) Recall

Recall, also commonly known as Sensitivity or True Positive Rate, is a vital performance metric utilized in the assessment of classification models. Its primary objective is to measure the model's ability to correctly identify all positive instances from the total actual positive instances present in the dataset. In essence, recall quantifies the proportion of true positive predictions relative to the total number of positive instances in the dataset.

Mathematically, recall is calculated as the ratio of true positives to the sum of true positives and false negatives. A high recall value indicates that the model has a lower rate of false negatives, signifying its proficiency in identifying a substantial portion of actual positive instances. On the other hand, a low recall value suggests a higher rate of false negatives, highlighting the model's tendency to overlook or misclassify positive instances as negative.

$$\text{Recall} = \frac{\text{True Positive(TP)}}{\text{True Positive(TP)} + \text{False Negative(FN)}}$$

Recall holds significant importance in applications where the detection of positive instances is of utmost priority, and the cost of missing positive instances (false negatives) is high. For instance, in medical diagnosis, a high recall model is essential to ensure the identification of actual cases of a disease, minimizing the risk of leaving undetected patients who require urgent treatment.

c) **F1 Score**

The F1 score, also known as the F-score or F-measure, is a crucial performance metric widely used in the evaluation of classification models. It seeks to strike a balance between precision and recall, providing a single comprehensive measure of a model's overall performance.

The F1 score is especially valuable when dealing with imbalanced datasets, where the number of positive and negative instances may vary significantly. In such cases, optimizing solely for precision or recall might lead to biased results. The F1 score addresses this issue by taking into account both precision and recall, offering a harmonic mean of the two metrics.

Mathematically, the F1 score is calculated as the harmonic mean of precision and recall, ensuring that both measures have equal impact on the final score. It is computed as the reciprocal of the sum of the reciprocals of precision and recall. As a result, the F1 score penalizes models that have significant disparities between precision and recall, thus encouraging a balanced performance.

$$\text{F1 Score} = \frac{2}{\frac{1}{Precision}+\frac{1}{Recall}}$$

$$\text{F1 Score} = \frac{TP}{TP+\frac{1}{2}(FP+FN)}$$

A high F1 score indicates that the model has achieved a good balance between precision and recall, demonstrating its capability to accurately classify positive instances while minimizing false positives and false negatives. On the other hand, a low F1 score suggests an imbalance in precision and recall, indicating areas where the model's performance may be improved.

The F1 score is a valuable metric for tasks where precision and recall are of equal importance, such as in medical testing, anomaly detection, and spam filtering. By incorporating both aspects of a model's performance, the F1 score provides a comprehensive assessment and aids researchers and practitioners in making informed decisions about the effectiveness of the classification model for specific applications.

d) Support

Support, in the context of performance evaluation for classification models, is a significant metric that indicates the number of occurrences or instances belonging to a particular class in a given dataset. It plays a crucial role in understanding the distribution of classes and assessing the relevance and representation of each class in the dataset.

The support value is calculated for each class individually and represents the count of actual instances present in the dataset belonging to that specific class. In essence, it reflects the prevalence or frequency of a class in the dataset and is instrumental in identifying class imbalances.

Class imbalances occur when some classes have a significantly higher number of instances than others, which can potentially lead to biased model performance. Support helps researchers and practitioners recognize these imbalances and take appropriate measures to address them. For instance, during model training, techniques like data augmentation, resampling, or class-weighted loss functions can be employed to mitigate the effects of imbalanced classes.

The support metric is especially useful in multi-class classification tasks, where there are multiple target classes. It provides insights into the relative representation of each class and helps identify classes that might have insufficient data for robust model training.

In summary, precision, recall, F1 score, and support are fundamental performance metrics in evaluating the performance of AI/ML models for classification tasks. They provide valuable insights into different aspects of the model's performance, such as accuracy, positive predictive power, sensitivity to positive instances, and class representation. By analysing these metrics, researchers and practitioners can fine-tune their models, address imbalances, and make informed decisions about the model's effectiveness and suitability for real-world applications.

4.4 Results

We have discussed the contents of the various datasets, our approach to classifying the samples of different classes contained in it, as well as the techniques and methods that we have employed in the proposed methodology. In this section, we will present the findings of the study. Following the described procedure, we set a classification model where the provided samples were first split into the training and testing samples respectively, and then were accordingly used to train and test our respective machine learning models. The result yields a higher classification accuracy of on the test samples when the Catboost algorithm is applied. Below table presents the classification accuracies on the various datasets of each of the selected machine learning algorithms.

Table 4.1: Classification accuracy of various boosting algorithms

MODEL → DATASETS ↓	ADABOOST	GRADIENT BOOSTING	XGBOOST	LIGHT GBM	CATBOOST
BREAST CANCER WISCONSIN	90.57	92	93.79	94	**96.91**
HEART ATTACK ANALYSIS & PREDICTION	74.70	76.59	80.11	83.1	**85.71**
LUNG CANCER	89.38	91.33	93.74	95.16	**96.47**
PIMA INDIANA DIABETES	69.32	70.56	72.07	74.79	**77.65**
DERMATALOGY	55.27	63.70	87.02	88.21	**91.47**
INDIAN LIVER PATIENT RECORDS	58.59	60.31	65.25	67.83	**71.69**
PARKINSON DISEASE	92.27	93.55	93.70	94.70	**95.69**

As seen in the results, the accuracy gradually increased from the traditional Adaboost algorithm to XGBoost algorithm, finally peaking at Catboost algorithm. These results mentioned in the table is basically the average percentage obtained on running the respective machine learning models for 20 times and then averaging their respective accuracy percentage obtained each time. Here, the train_test_split is basically considered to be 0.3 (i.e., from the entire dataset, 70% is training samples and 30% is test samples).

Apart from the classification accuracy, the standard deviation is another useful tool to judge the performance of the classification model. The standard deviation plays a crucial role in analysing the performance of a machine learning model, especially when evaluating its predictive capability and generalization ability. It is commonly used as a measure of the model's variability or dispersion in its predictions.

When assessing the performance of a machine learning model, we often split the available data into training and testing sets. The model is trained on the training set and then evaluated on the testing set to understand how well it can generalize to new, unseen data. During this process, multiple iterations or runs of the model may be performed to account for randomness and ensure a more reliable assessment. The standard deviation comes into play when we run the model multiple times and collect the evaluation metrics (e.g., accuracy, precision, recall, F1-score) for each run. These metrics may vary slightly between different runs due to the randomness involved in model training and data splitting. Below table presents the standard deviation on the various datasets of each of the selected machine learning algorithms.

Table 4.2: Standard Deviation of the accuracy values obtained using various boosting algorithms

MODEL ➞ DATASETS ↓	**ADABOOST**	**GRADIENT BOOSTING**	**XGBOOST**	**LIGHT GBM**	**CATBOOST**
BREAST CANCER WISCONSIN	0.89	0.70	0.54	0.50	**0.25**
HEART ATTACK ANALYSIS & PREDICTION	5.96	3.59	2.43	2.50	**1.22**
LUNG CANCER	2.56	2.50	1.93	1.19	**1.03**
PIMA INDIANA DIABETES	1.56	1.39	1.24	1.09	**0.90**
DERMATALOGY	12.59	3.50	2.18	2.50	**1.91**
INDIAN LIVER PATIENT RECORDS	5.14	3.10	2.02	1.95	**1.45**
PARKINSON DISEASE	7.3	4.12	3.5	2.6	**1.7**

By calculating the standard deviation of the evaluation metrics, we can get a sense of how consistent the model's performance is across different runs. A lower standard deviation indicates that the model's predictions are more consistent, whereas a higher standard deviation suggests that the performance can vary significantly between different runs.

From the above table, it can be realised that the Catboost model's performance is more consistent than the other two models, across different subsets of data. This is a positive sign, indicating that the model is likely to generalize well to new, unseen data.

Besides, the confusion matrix of classification is another useful tool to judge the performance of the classification model. The confusion matrix provides more details on the output of the classification process. It tells us how many samples of each class were tested and how many of them were classified correctly based on the model. In the best-case scenario (i.e., when the classification accuracy would be 100%) only the diagonal boxes of the matrix would contain non-zero values or the number of tested samples of the corresponding class, and all the other boxes would contain zeros.

Precision, recall, F1-score and Support are other four commonly used parameters to measure the potency of a classifier.

To put the outcome of our classification model in context, we have compared our acquired results with that of other similar studies involving the selected datasets in Table I. It shows that not only in terms of classification accuracy but also in terms of standard deviation, precision, recall, F1 score, and support, the proposed method, i.e., the Catboost algorithm outperforms the other methods considered i.e., Adaboost, Gradient Boost, XGBoost and LighTGBM algorithms respectively.

Table 4.3: Precision scores obtained using various boosting algorithms

MODEL ➞ DATASETS ↓	ADABOOST	GRADIENT BOOSTING	XGBOOST	LIGHT GBM	CATBOOST
BREAST CANCER WISCONSIN	91	92	95	96	**98**
HEART ATTACK ANALYSIS & PREDICTION	75	79	82	84	**85**
LUNG CANCER	88	90	92	93	**95**
PIMA INDIANA DIABETES	67	69	72	73	**75**
DERMATALOGY	56	59	65	74	**79**
INDIAN LIVER PATIENT RECORDS	60	63	64	69	**70**
PARKINSON DISEASE	84	87	90	93	**94**

From the above table, it can be seen that CatBoost model is making fewer false positive predictions compared to other algorithms. It is more accurate at identifying positive instances without mistakenly labelling negative instances as positive.

Table 4.4: Recall scores obtained using various boosting algorithms

MODEL ➞ DATASETS ↓	ADABOOST	GRADIENT BOOSTING	XGBOOST	LIGHT GBM	CATBOOST
BREAST CANCER WISCONSIN	91	92	95	96	**98**
HEART ATTACK ANALYSIS & PREDICTION	75	79	82	84	**85**
LUNG CANCER	88	90	92	93	**95**
PIMA INDIANA DIABETES	67	69	72	73	**75**
DERMATALOGY	57	59	66	79	**85**
INDIAN LIVER PATIENT RECORDS	62	65	74	75	**78**
PARKINSON DISEASE	86	89	90	92	**93**

The above table suggests that the CatBoost model is better at capturing a larger proportion of the actual positive instances in the dataset. It is sensitive to positive cases and tends to miss fewer of them compared to other algorithms.

Table 4.5: F1- scores obtained using various boosting algorithms

MODEL ➞ DATASETS ↓	ADABOOST	GRADIENT BOOSTING	XGBOOST	LIGHT GBM	CATBOOST
BREAST CANCER WISCONSIN	91	92	95	96	**98**
HEART ATTACK ANALYSIS & PREDICTION	77	79	82	83	**85**
LUNG CANCER	88	91	92	93	**96**
PIMA INDIANA DIABETES	71	74	76	77	**80**
DERMATALOGY	57	60	66	72	**85**
INDIAN LIVER PATIENT RECORDS	65	69	73	75	**77**
PARKINSON DISEASE	85	89	90	93	**95**

The F1-score combines precision and recall into a single metric. A higher F1-score indicates that your CatBoost model achieves a better balance between precision and recall than other algorithms. It is providing strong overall performance by minimizing both false positives and false negatives.

Table 4.6: Support scores obtained using various boosting algorithms

MODEL ➞ DATASETS ↓	ADABOOST	GRADIENT BOOSTING	XGBOOST	LIGHT GBM	CATBOOST
BREAST CANCER WISCONSIN	150	155	157	168	**171**
HEART ATTACK ANALYSIS & PREDICTION	79	82	85	90	**91**
LUNG CANCER	132	137	140	142	**143**
PIMA INDIANA DIABETES	142	145	150	152	**154**
DERMATALOGY	57	62	66	79	**85**
INDIAN LIVER PATIENT RECORDS	123	146	157	167	**175**
PARKINSON DISEASE	41	43	44	50	**59**

A higher support value means that your CatBoost model is dealing with a dataset where the distribution of classes is well-represented. This indicates that the model is tested on a substantial number of data points for each class, providing robust results.

In summary, CatBoost model consistently outperforms other boosting algorithms in terms of precision, recall, F1-score, and support, which is a strong indication that CatBoost is a suitable choice for classification task. It excels in correctly identifying positive instances, minimizing false positives, and achieving a good balance between precision and recall.

CHAPTER 5

Conclusion

This study presents an empirical analysis of five widely used variants of boosting algorithms: Adaboost, Gradient Boosting, XGBoost, LightGBM and CatBoost, all of which are acknowledged for their effectiveness and precision as models. Specifically, the study juxtaposes the performance of CatBoost, particularly in terms of accuracy and several other key performance metrics, with the performance exhibited by the other boosting algorithms across a diverse array of classification tasks.

As we conclude this review, it is clear that CatBoost's capabilities hold the promise of enhancing the accuracy and efficacy of machine learning models in a wide array of applications. Researchers and practitioners alike can confidently consider CatBoost as a valuable addition to their arsenal in the pursuit of accurate and reliable classification solutions.

References:

[1] Yoav Freund, Robert E. Schapire, “Experiments with a new boosting algorithm.” Jan. 22, 1996. [Online]. Available: chrome-extension://efaidnbmnnnibpcajpcglclefindmkaj/https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=d186abec952c4348870a73640bf849af9727f5a4

[2] Y. Freund and R. Schapire, “Game Theory, On-line Prediction and Boosting” Nov. 2000, doi: 10.1145/238061.238163.

[3] Y. Freund and R. E. Schapire, “A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting,” Journal of Computer and System Sciences, vol. 55, no. 1, pp. 119–139, Aug. 1997, doi: 10.1006/jcss.1997.1504.

[4] T. Chengsheng, X. Bing, and L. Huacheng, “The Application of the AdaBoost Algorithm in the Text Classification,” in 2018 2nd IEEE Advanced Information Management,Communicates,Electronic and Automation Control Conference (IMCEC), May 2018, pp. 1792–1796. doi: 10.1109/IMCEC.2018.8469497.

[5] J. Thongkam, G. Xu, Y. Zhang, and F. Huang, “Breast Cancer Survivability via AdaBoost Algorithms,” The Proceedings of the second Australasian workshop on Health data and knowledge management Wollongong, pp. 55–64, Jan. 2008.

[6] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” in Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001, Dec. 2001, p. I–I. doi: 10.1109/CVPR.2001.990517.

[7] J. Friedman, T. Hastie, and R. Tibshirani, “Additive logistic regression: a statistical view of boosting (With discussion and a rejoinder by the authors),” The Annals of Statistics, vol. 28, no. 2, pp. 337–407, Apr. 2000, doi: 10.1214/aos/1016218223.

[8] J. Tanha, M. van Someren, and H. Afsarmanesh, “An AdaBoost Algorithm for Multiclass Semi-supervised Learning,” in 2012 IEEE 12th International Conference on Data Mining, Dec. 2012, pp. 1116–1121. doi: 10.1109/ICDM.2012.119.

[9] T. Hastie, S. Rosset, J. Zhu, and H. Zou, “Multi-class AdaBoost,” Statistics and Its Interface, vol. 2, no. 3, pp. 349–360, 2009, doi: 10.4310/SII.2009.v2.n3.a8.

[10] R. Lienhart and J. Maydt, “An Extended Set of Haar-like Features for Rapid Object Detection,” presented at the Proceedings of the International Conference on Image Processing, Feb. 2002, p. I–900. doi: 10.1109/ICIP.2002.1038171.

[11] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, and F. Herrera, “A Review on Ensembles for the Class Imbalance Problem: Bagging-, Boosting-, and Hybrid-Based Approaches,” IEEE Trans. Syst. Man Cybern. Part C Appl. Rev., vol. 42, no. 4, pp. 463–484, Jul. 2012, doi: 10.1109/TSMCC.2011.2161285.

[12] P. Bühlmann and T. Hothorn, “Boosting Algorithms: Regularization, Prediction and Model Fitting,” Statistical Science, vol. 22, May 2008, doi: 10.1214/07-STS242.

[13] T. Chen and C. Guestrin, “XGBoost: A Scalable Tree Boosting System,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery

and Data Mining, in KDD ’16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[14] T. Hastie, R. Tibshirani, and J. Friedman, “Boosting and Additive Trees,” in The Elements of Statistical Learning: Data Mining, Inference, and Prediction, T. Hastie, R. Tibshirani, and J. Friedman, Eds., in Springer Series in Statistics. New York, NY: Springer, 2009, pp. 337–387. doi: 10.1007/978-0-387-84858-7_10.

[15] J. T. Leonard and K. Roy, “QSAR modeling of anti-HIV activities of alkenyldiarylmethanes using topological and physicochemical descriptors,” Drug Design and Discovery, vol. 18, no. 4, pp. 165–180, 2003.

[16] I. Karabayir, S. M. Goldman, S. Pappu, and O. Akbilgic, “Gradient boosting for Parkinson’s disease diagnosis from voice recordings,” BMC Medical Informatics and Decision Making, vol. 20, no. 1, p. 228, Sep. 2020, doi: 10.1186/s12911-020-01250-7.

[17] S. Bakas et al., “GLISTRboost: Combining Multimodal MRI Segmentation, Registration, and Biophysical Tumor Growth Modeling with Gradient Boosting Machines for Glioma Segmentation,” in Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries, A. Crimi, B. Menze, O. Maier, M. Reyes, and H. Handels, Eds., in Lecture Notes in Computer Science. Cham: Springer International Publishing, 2016, pp. 144–155. doi: 10.1007/978-3-319-30858-6_13.

[18] S. Lundberg and S.-I. Lee, “A Unified Approach to Interpreting Model Predictions,” Dec. 2017.

[19] R. Y. Gupta, S. Sai Mudigonda, P. K. Kandala, and P. K. Baruah, “Implementation of a Predictive Model for Fraud Detection in Motor Insurance using Gradient Boosting Method and Validation with Actuarial Models,” in 2019 IEEE International Conference on Clean Energy and Energy Efficient Electronics Circuit for Sustainable Development (INCCES), Dec. 2019, pp. 1–6. doi: 10.1109/INCCES47820.2019.9167733.

[20] A. Ogunleye and Q.-G. Wang, “XGBoost Model for Chronic Kidney Disease Diagnosis,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 17, no. 6, pp. 2131–2140, Nov. 2020, doi: 10.1109/TCBB.2019.2911071.

[21] M. Yang, L. Song, J. Xu, C. Li, and G. Tan, “The Tradeoff Between Privacy and Accuracy in Anomaly Detection Using Federated XGBoost.” arXiv, Oct. 14, 2019. doi: 10.48550/arXiv.1907.07157.

[22] C. Zhang et al., “Fraud Detection under Multi-Sourced Extremely Noisy Annotations,” in Proceedings of the 30th ACM International Conference on Information & Knowledge Management, in CIKM ’21. New York, NY, USA: Association for Computing Machinery, Oct. 2021, pp. 2497–2506. doi: 10.1145/3459637.3482433.

[23] Z. Wu, T. Lei, C. Shen, Z. Wang, D. Cao, and T. Hou, “ADMET Evaluation in Drug Discovery. 19. Reliable Prediction of Human Cytochrome P450 Inhibition Using Artificial Intelligence Approaches,” J. Chem. Inf. Model., vol. 59, no. 11, pp. 4587–4601, Nov. 2019, doi: 10.1021/acs.jcim.9b00801.

[24] M. Su et al., “Genomic analysis of variability in Delta-toxin levels between Staphylococcus aureus strains,” PeerJ, vol. 8, p. e8717, Mar. 2020, doi: 10.7717/peerj.8717.

[25] G. Ke et al., “LightGBM: A Highly Efficient Gradient Boosting Decision Tree,” in Advances in Neural Information Processing Systems, Curran Associates, Inc., 2017.

Accessed: Sep. 03, 2023. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/6449f44a102fde848669bdd9eb6b76fa-Abstract.html

[26] K. Guo and L. Zhang, "Adaptive multi-objective optimization for emergency evacuation at metro stations," Reliability Engineering & System Safety, vol. 219, p. 108210, Mar. 2022, doi: 10.1016/j.ress.2021.108210.

[27] L. Kopitar, P. Kocbek, L. Cilar, A. Sheikh, and G. Stiglic, "Early detection of type 2 diabetes mellitus using machine learning-based prediction models," Scientific Reports, vol. 10, no. 1, Art. no. 1, Jul. 2020, doi: 10.1038/s41598-020-68771-z.

[28] G. Csizmadia, K. Liszkai-Peres, B. Ferdinandy, Á. Miklósi, and V. Konok, "Human activity recognition of children with wearable devices using LightGBM machine learning," Scientific Reports, vol. 12, no. 1, Art. no. 1, Mar. 2022, doi: 10.1038/s41598-022-09521-1.

[29] G. Yang, S. Du, Q. Duan, and J. Su, "A Novel Data-Driven Method for Medium-Term Power Consumption Forecasting Based on Transformer-LightGBM," Mobile Information Systems, vol. 2022, p. e5465322, Aug. 2022, doi: 10.1155/2022/5465322.

[30] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, "CatBoost: unbiased boosting with categorical features." arXiv, Jan. 20, 2019. doi: 10.48550/arXiv.1706.09516.

[31] P. S. Kumar, A. K. K, S. Mohapatra, B. Naik, J. Nayak, and M. Mishra, "CatBoost Ensemble Approach for Diabetes Risk Prediction at Early Stages," in 2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology(ODICON), Jan. 2021, pp. 1–6. doi: 10.1109/ODICON50556.2021.9428943.

[32] J. N. M. Avelino, E. P. Felizmenio Jr., and P. C. Naval Jr., "Unraveling COVID-19 Misinformation with Latent Dirichlet Allocation and CatBoost," in Advances in Computational Collective Intelligence, C. Bădică, J. Treur, D. Benslimane, B. Hnatkowska, and M. Krótkiewicz, Eds., in Communications in Computer and Information Science. Cham: Springer International Publishing, 2022, pp. 16–28. doi: 10.1007/978-3-031-16210-7_2.

[33] A. Samat, E. Li, P. Du, S. Liu, and J. Xia, "GPU-Accelerated CatBoost-Forest for Hyperspectral Image Classification Via Parallelized mRMR Ensemble Subspace Feature Selection," IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens., vol. 14, pp. 3200–3214, 2021, doi: 10.1109/JSTARS.2021.3063507.

[34] A. Samat, E. Li, P. Du, S. Liu, Z. Miao, and W. Zhang, "CatBoost for RS Image Classification With Pseudo Label Support From Neighbor Patches-Based Clustering," IEEE Geoscience and Remote Sensing Letters, vol. 19, pp. 1–5, 2022, doi: 10.1109/LGRS.2020.3038771.

[35] Y. Bo, Q. Liu, X. Huang, and Y. Pan, "Real-time hard-rock tunnel prediction model for rock mass classification using CatBoost integrated with Sequential Model-Based Optimization," Tunnelling and Underground Space Technology, vol. 124, p. 104448, Jun. 2022, doi: 10.1016/j.tust.2022.104448.

[36] O. M. William Wolberg, "Breast Cancer Wisconsin (Diagnostic)." UCI Machine Learning Repository, 1993. doi: 10.24432/C5DW2B.

[37] Wisconsin Breast Cancer Dataset (UW CS ftp server) (https://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/)

[38] UCI Machine Learning Repository (https://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/)

[39] Heart Attack Analysis dataset (https://www.kaggle.com/datasets/rashikrahmanpritom/heart-attack-analysis- prediction-dataset)

[40] Lung Cancer dataset (https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer)

[41] Diabetes datset in UCI Repository(https://www.kaggle.com/datasets/mysarahmadbhat/lung-cancer)

[42] Diabetes dataset in Kaggle (https://www.kaggle.com/datasets/uciml/pima-indians-diabetes-database)

[43] Dermatology dataset in OpenML (https://www.openml.org/search?type=data&sort=runs&id=35&status=active)

[44] Dermatology dataset on Kaggle (https://www.kaggle.com/datasets/syslogg/dermatology-dataset)

[45] Indian Liver patient dataset (http://archive.ics.uci.edu/dataset/225/ilpd+indian+liver+patient+dataset)

[46] Indian Liver patient dataset on Kaggle (https://www.kaggle.com/datasets/uciml/indian-liver-patient-records)

[47] Parkinson Disease Detection dataset- UCI repository (https://archive.ics.uci.edu/dataset/174/parkinsons)

[48] Parkinson Disease Detection dataset- Kaggle (https://www.kaggle.com/datasets/debasisdotcom/parkinson-disease-detection)