REPUBLIC OF TÜRKİYE

ALTINBAŞ UNIVERSITY

Institute of Graduate Studies

Information Technologies

PREDICTING USER CLICKS ON ONLINE ADVERTISEMENTS USING MACHINE LEARNING

Ashraf Farhan Hatem AL-KHAFAJI

Master's Thesis

Supervisor

Asst. Prof. Dr. Oğuz KARAN

İstanbul, 2023

PREDICTING USER CLICKS ON ONLINE ADVERTISEMENTS USING MACHINE LEARNING

Ashraf Farhan Hatem AL-KHAFAJI

Information Technologies

Master's Thesis

ALTINBAŞ UNIVERSITY

2023

The thesis/dissertation titled PREDICTING USER CLICKS ON ONLINE ADVERTISEMENTS USING MACHINE LEARNING prepared by ASHRAF FARHAN HATEM AL-KHAFAJI and submitted on …… (DATE) has been **accepted unanimously** for the degree of Master of Science in Information Technologies

Asst. Prof. Dr. Oğuz KARAN

Supervisor

Thesis Defense Committee Members:

Academic Title - First/Last Name	Department, University	_____
Academic Title - First/Last Name	Department, University	_____
Academic Title - First/Last Name	Department, University	_____
Academic Title - First/Last Name	Department, University	_____

I hereby declare that this thesis meets all format and submission requirements of a Master's thesis.

Submission date of the thesis to Institute of Graduate Studies: ___/___/___

I hereby declare that all information/data presented in this graduation project has been obtained in full accordance with academic rules and ethical conduct. I also declare all unoriginal materials and conclusions have been cited in the text and all references mentioned in the Reference List have been cited in the text, and vice versa as required by the abovementioned rules and conduct.

Ashraf Farhan Hatem AL-KHAFAJI

Signature

ABSTRACT

PREDICTING USER CLICKS ON ONLINE ADVERTISEMENTS USING MACHINE LEARNING

AL-KHAFAJI, Ashraf Farhan Hatem
M.Sc., Information Technologies, Altınbaş University,
Supervisor: Asst. Prof. Dr. Oğuz KARAN
Date: 12 / 2013
Pages: 79

This thesis explores the detection of user behavior within online advertising to better understand user preferences and refine ad strategies. As online ads are instrumental in reaching broad audiences, targeted approaches are vital for enhancing campaign effectiveness. The research employs data analysis, preprocessing, and machine learning (ML) techniques on a dataset detailing user behaviors like browsing history, ad clicks, and demographics. ML methods, including random forest, GB, and LR, are utilized alongside XAI tools like LIME and SHAP to highlight feature importance. The study aims to discern user behavior patterns to improve ad targeting. Models are further optimized using parameter tuning, and ensemble strategies, such as soft and hard voting, are used to aggregate individual predictions, boosting detection accuracy. Performance metrics, ranging from accuracy to specificity, are used to assess model efficacy. Results show that ensemble methods, particularly soft voting, outperform other techniques in accuracy.

Keywords: User Behavior Detection, Online Advertising, Machine Learning, Parameter Tuning Ensemble Voting, XAI.

TABLE OF CONTENTS

Pages

LIST OF FIGURES

Pages

ABBREVIATIONS

ML	:	Artificial Intelligence
AI	:	Artificial Intelligence
XAI	:	Explainable Artificial Intelligence
DT	:	Decision Tree
SL	:	Supervised learning
USL	:	Unsupervised learning
SSL	:	Semi Supervised Learning
RL	:	Reinforcement Learning
EL	:	Ensemble Learning
RF	:	Random Forest
GB	:	Gradient Boosting
LR	:	Logistic Regression
KNN	:	K-Nearest Neighbors
LIME	:	Local Interpretable Model-Agnostic Explanations
SHAP	:	SHapley Additive ExPlanations
SVM	:	Support Vector Machines
ACC	:	Accuracy
PRE	:	Precision
REC	:	Recall
F1-S	:	F1-Score
CR	:	Classification Report
CM	:	Confusion Matrix

1. INTRODUCTION

1.1 OVERVIEW

A website visitor is defined as someone who visits a website or social media platform and interacts with its content in the context of online advertising [1]. These signify a new space where individuals, groups, and even governments can engage in social, political, commercial, and educational interactions as well as information exchange [2]. Users can also browse various pages, click on links, look up products, and spend time on the website. As a result, businesses all over the world have begun to consider how utilizing these platforms could aid in drawing clients and developing lucrative marketing relationships with those clients. Therefore, analysing their actions and interactions in greater detail is necessary to gain valuable insights into why they behave as they do and it is essential for businesses to develop effective advertising campaigns that aim for the appropriate audience and motivate users to interact with their content [3]. It enables businesses to recognize patterns and trends, allowing them to optimize their advertising strategies accordingly. In recent years, the rise of digital advertising has transformed the way businesses reach their target audiences. However, with the vast amount of advertising content available online, users can easily become overwhelmed and may choose not to engage with ads. The significant amount of money that organizations spend on advertising campaigns indicates the great interest in this field. For example, in 2016, Statista [4] reported that approximately 524.58 billion USD was invested in advertising campaigns. Similarly, social media ads have also gained significant attention, with around 32.3 billion USD spent on desktop and mobile social media ads in 2016, as stated by Statista [5]. However, this leads to the question of whether these campaigns are feasible from the perspective of the company. Additionally, marketers constantly face the challenge of designing social media ads that are more effective and appealing to potential customers. Facebook accounted for 71.8% of all visitors to social media sites in the United States as of May 2021, making it the most popular social media platform [6]. Twitter came in second with 9.15%, followed by Instagram and Pinterest with 3.82 % and 12.4 %, respectively [7]. The meta’s Family of Apps annual revenue was 117.92 billion US dollars in 2021, an increase from 85.97 billion in 2020. The company's annual revenue has climbed by over 114 billion dollars in the last ten years [8].

User behaviour can be classified into two categories: click behaviour and non-click behaviour [9]. Click behaviour refers to the action of clicking on an advertisement. It is an essential metric for measuring the success of an advertising campaign. Higher click rates indicate that the ad content is relevant to the user and effectively captures their attention. Non-click behaviour refers to the behaviour of users who do not click on an advertisement. It can provide insights into user preferences and can be used to improve the effectiveness of advertising campaigns. Analyzing non-click behaviour can reveal patterns in user behaviour, such as preferences for certain types of ad content or specific times of day for viewing ads.

1.2 PROBLEM STATEMENT

Effective ads are critical for the success of advertising campaigns. To achieve this, advertisers must understand their target audience's behavior and preferences on digital platforms. However, traditional methods of analyzing user behavior, such as surveys or focus groups, are often time-consuming and ineffective. These methods may also suffer from selection bias, where the results may not accurately represent the target audience as a whole. Furthermore, the rise of ad-blockers has made it even more challenging for advertisers to reach their target audience. Ad-blockers prevent ads from being displayed, making it difficult for advertisers to collect data on user behavior and preferences. User behavior detection can overcome this challenge by analyzing data from various sources, such as website analytics or social media activity.

In addition to creating effective ads, user behavior detection can help prevent click fraud and other fraudulent activities that can harm ad campaigns' performance. By identifying and preventing such activities, advertisers can ensure that their ads are reaching their intended audience, and their ads spend is being used efficiently. As the digital advertising landscape continues to evolve, with new platforms and technologies emerging, advertisers must stay ahead of the curve. User behavior detection can help advertisers identify emerging trends and preferences, enabling them to adapt their advertising strategies and stay relevant.

1.3 RESEARCH MOTIVATION

The motivation for studying ads user behavior detection is the potential benefits for both advertisers and users. By creating personalized and targeted ads, advertisers can improve

the user experience, increase engagement, and ultimately drive conversions, leading to a higher return on investment. At the same time, users are more likely to engage with ads that are relevant and personalized, leading to a more positive user experience. Additionally, the development of an accurate and efficient ML model for user behavior detection can contribute to the advancement of the field of ML and artificial intelligence (AI), as well as the digital advertising industry as a whole.

1.4 RESEARCH OBJECTIVES

The research objective in studying ads user behavior detection is to devise a precise and efficient ML model capable of real-time analysis of user behavior and preferences, thereby enabling the creation of bespoke ads tailored to each user's unique behavior and preferences. The model is expected to manage vast data quantities and intricate algorithms, offering instantaneous assessment and ad optimization rooted in user behavior. Furthermore, by harnessing explainable artificial intelligence (XAI), the model aims to elucidate its decision-making process, ensuring transparency in ad targeting. This not only enhances user experience by serving pertinent and captivating ads but also bolsters the return on investment for advertisers by fostering conversions via bespoke ad targeting. The overarching aspiration is to bolster advancements in ML and AI fields, culminating in a precise, efficient, and transparent ML model for user behavior detection.

1.5 RESEARCH QUESTIONS

a. How can user behavior detection be used to improve the user experience and increase the return on investment for advertisers?
b. How can ML be used to accurately and efficiently analyze user behavior and preferences in relation to digital advertising?
c. How can an ML model be developed that is capable of handling large amounts of data and complex algorithms, providing real-time analysis and optimization of ads based on user behavior?
d. hat impact do personalized and targeted ads have on user engagement and conversion rates, and how can this impact be measured?
e. What are the most effective methods for measuring the effectiveness of personalized and targeted ads created using user behavior detection techniques?

1.6 RESEARCH CONTRIBUTION

The contribution of this research is:

a. The formulation of an ML-based methodology to scrutinize user behavior and preferences concerning digital advertising, complemented by the optimization of advertising initiatives via EL methodologies.
b. Through the employment of ML algorithms such as RF, LR, and GB, and the integration of EL techniques like VotingClassifier, this research endeavors to craft ads that are both impactful and align with the inclinations of the target demographic, fostering higher conversion rates.
c. This investigation extends the frontiers of knowledge in the realm of tailored and pinpointed advertising by identifying nascent trends and inclinations in user behavior, and appraising the efficacy of such tailored advertising strategies.
d. A significant highlight of this research is the integration of XAI to ensure that the ML models are not only effective but also transparent, allowing stakeholders to comprehend and trust the underlying decision-making processes of the models.
e. The ultimate aspiration is to elevate the user experience and augment the return on investment for marketers by orchestrating advertising campaigns that are more precise, agile, and resonate with their intended audience.

1.7 RESEARCH PLAN

The research blueprint for this thesis unfolds in a structured manner.

Literature Review: the study will engage with prior scholarly work on user behavior in online advertising, exploring various ad modalities. It will also dissect various user behaviors and their impact on advertising while diving deep into the domain of click advertising. The role of ML and AI in enhancing ad strategies will be thoroughly examined, focusing on specific algorithms and ensemble techniques.

Methodology: the research design will be unveiled, detailing the dataset sourcing, its analytical framework, preprocessing, and the classifiers being adopted, with an emphasis on ensemble methods.

Results and Discussion: will introduce the evaluative framework, tools, and libraries, followed by performance assessments of models, both pre and post-optimization. The section will culminate in a comparative analysis of ensemble techniques.

Conclusion and Future Work: encapsulating the core findings, their implications, and prospective research avenues in the domain.

2. LITERATURE REVIEW

2.1 RELATED WORKS

A review of the literature on user behavior in ML refers to an assessment of the previous studies conducted on this subject. The review aims to examine the utilization of ML algorithms in modeling and comprehending user behavior, and will analyze the principal discoveries and patterns identified in this field of study.

This study [10] looks into how big data technologies may be used to forecast customer behavior on social media sites. The study use mathematical modeling created by ML to anticipate customer behavior by examining consumer activity on social media using numerous characteristics and criteria. Pre-processing methods for the data include outlier detection, noise reduction, error detection, and duplicate record removal. The decision tree (DT) model is the one that predicts consumer behavior on social media sites most accurately, according to the results. According to the survey, customer deviation varies among social media platforms from 12.22% to 99.51%. The largest root mean square error is 156556.45293 and the lowest is 20691.7870. The model's ACC varies between 0.02238 and 0.98292.

This paper [11] focuses on demographic targeting through LR models, utilizing variables derived from user affinities for webpages and online behavior. The proposed models include a binomial logistic model for predicting gender and a multinomial logistic model for estimating age categories. Only variables with significant influence, as determined by stepwise regression, are used. The impact of factors on areas of interest for women, men, and specific age categories are quantified using odds ratios. The models are evaluated using classification measures derived from a CM, and their ACC is validated using a large dataset from a real advertising campaign. The study demonstrates the usefulness of logistic models for estimating the probabilities of an internet user belonging to a target group.

This paper [12] developed a method for determining whether each click is suspicious by comparing it to a number of criteria, which would eliminate click fraud in ad networks. The system, which included three major agents, performed well in a testing setting at identifying different forms of assaults. To appropriately categorize assaults, the authors recommended that the weights of the criteria be improved in a real-world situation. The

technique offers a potential remedy for the click fraud issue that plagues internet advertising and can help shield marketers from monetary losses.

In order to develop a comprehensive contextual advertising strategy, this study investigates ways to enhance company page targeting on Facebook using consumer behavior targeting, age, and gender. This research [13] suggests an algorithm that uses DTs to regulate and optimize a company's marketing strategy by fusing statistical training principles with commercial objectives. The algorithm's goal is to maximize campaign revenue by simulating consumer engagement with a company page on Facebook. The suggested technique was evaluated using data from a Facebook contextual advertising campaign, and the simulation outcomes reveal that the algorithm suggests target groups that are much more profitable than industry standards. The study supports the utility of the suggested approach, which develops DTs for customer engagement while taking company objectives into consideration.

The approach suggested in this study employs association rule mining and customer feedback categorization to suggest projects to freelancers. To categorize completed works as good or negative, they begin by compiling the work histories of freelancers and utilizing LR and Linear SVM models to assess the mood of customer comments. Finally, we utilize association rule mining to determine which skill sets freelancers frequently employ across both types of successfully completed assignments. We then employ set operations to find employment that correspond to the positive frequent skillsets while eliminating those that do not. Lastly, to produce a more precise suggestion, we use a collaborative filtering algorithm that takes into account client ratings, minimum budget/hourly rate, deadline, and re-hire. Our tests on actual datasets from several online marketplaces show that the ACC of our suggested method's recommendations for acceptable jobs is 83.40% (LR) and 84.03% (Linear SVM).

In the research presented in this paper [9], the Search-based Interest Model (SIM) is introduced, employing two distinct search units: the General Search Unit (GSU) and the Exact Search Unit (ESU). These units collaboratively work to derive user interests from extensive sequential behavior data. The GSU initially performs a broad search using query information from a candidate item to yield a relevant Sub User Behavior Sequence (SBS). Subsequently, the ESU precisely models the relationship between the SBS and the

candidate item, enhancing SIM's ability to accurately and efficiently capture lifetime sequential behavior data. This novel approach has been effectively integrated into Alibaba's display advertising system since 2019, resulting in notable improvements of 7.1% in click-through rate (CTR) and 4.4% in revenue per mille (RPM). Currently, SIM manages the bulk of traffic in Alibaba's operational system.

This essay focuses on the problem of click fraud in internet advertising, which is dangerous for e-commerce and the advertising sector [14]. Although there have been improvements made to traffic filtering methods, efficient fraud detection algorithms are still required to safeguard online advertising firms. In order to discover IP addresses that generate plenty of hits but never lead to app installs, the study analyzes click patterns across a dataset of 200 million clicks over four days. The ACC was 98% because to the implementation of the LightGBM ML algorithm, a technique similar to GB and DT. The research emphasizes the significance of creating efficient fraud detection algorithms for online advertising organizations and relies on a literature analysis to verify the results.

2.2 USER BEHAVIOUR

Advertising is an essential part of the modern consumer economy, and as such, understanding user behavior in relation to advertising is crucial for marketers and advertisers. User behavior refers to the actions, attitudes, and responses of individuals when exposed to advertising [15]. This includes factors such as attention, perception, trust, and action. In this section, we will explore various types of user behavior in advertising, including how users engage with ads, their attitudes towards them, and the actions they take in response. We will also examine how cross-cultural and demographic differences influence user behavior, as well as emerging trends and implications for advertisers and policymakers. By understanding user behavior in advertising, marketers and advertisers can develop more effective advertising strategies that resonate with their target audience and drive better outcomes. Additionally, policymakers can use this information to regulate advertising practices and protect consumers from deceptive or harmful ads.

2.2.1 Types of Online Advertising

2.2.1.1 Search advertising

Also known as search engine marketing (SEM), is a form of online advertising that involves displaying text ads within search engine results pages, such as Google Ads [16]. These ads are triggered by specific keywords that users search for and are designed to appear at the top or bottom of the search results page. The ads are usually marked as "sponsored" or "ad" to distinguish them from organic search results. It is highly targeted and allows businesses to reach users who are actively searching for products or services related to their business. By bidding on specific keywords, businesses can control when and where their ads appear, and can target users based on geographic location, device type, and other factors. This advertising type can be an effective way for businesses to drive traffic to their website and generate leads or sales. However, it can also be highly competitive, and businesses must carefully manage their bids and budgets to ensure they are getting the most value for their advertising spend. Additionally, search advertising requires ongoing optimization and monitoring to ensure the ads are performing as expected and to make adjustments as needed.

Search Engine Optimization (SEO) is another important aspect of online advertising that involves optimizing a website's content and structure to rank higher in search engine results pages (SERPs) organically [17]. SEO aims to increase a website's visibility and attract more organic traffic by targeting specific keywords and improving the website's overall user experience. SEO involves various techniques, such as optimizing meta tags and descriptions, improving website speed and mobile-friendliness, creating high-quality content, and building backlinks [18]. By improving a website's SEO, businesses can increase their visibility and attract more traffic, without having to pay for advertising. However, SEO requires ongoing effort and can take time to see results, making it a long-term strategy. Integrating SEO with search advertising can create a more comprehensive online advertising strategy, allowing businesses to reach users through both paid and organic search results [18].

2.2.1.2 Social media advertising

Sponsored posts or display adverts that appear on social media sites like Facebook, Instagram, and Twitter are examples of social media advertising [19], [20]. By targeting

these advertisements to particular demographics, interests, and behaviors, companies can connect with those who are most likely to be interested in their goods or services [21]. There are several ways that social media advertising can be used: display ads, video ads, carousel ads, sponsored posts, and more [22]. These advertisements may show up on users' newsfeeds or in other sections of the website, including the stories or sidebars. To set them apart from organic material, they are frequently labeled as "sponsored" or "ad". The opportunity to communicate with visitors through interactive material, such surveys, quizzes, and competitions, is one advantage of this kind of advertising. This can assist companies in increasing sales, lead generation, and brand recognition [23]. Furthermore, social media advertising enables companies to monitor the effectiveness of their ads in real-time and modify them as necessary to enhance their outcomes [20]. Social media advertising is not without its difficulties, though. Companies must produce engaging material that appeals to their target market, and there is a risk of overexposure to ads on the site and ad fatigue. To make sure they are getting the most out of their advertising budget and achieving the intended outcomes, businesses need to closely monitor and optimize their social media advertising efforts.

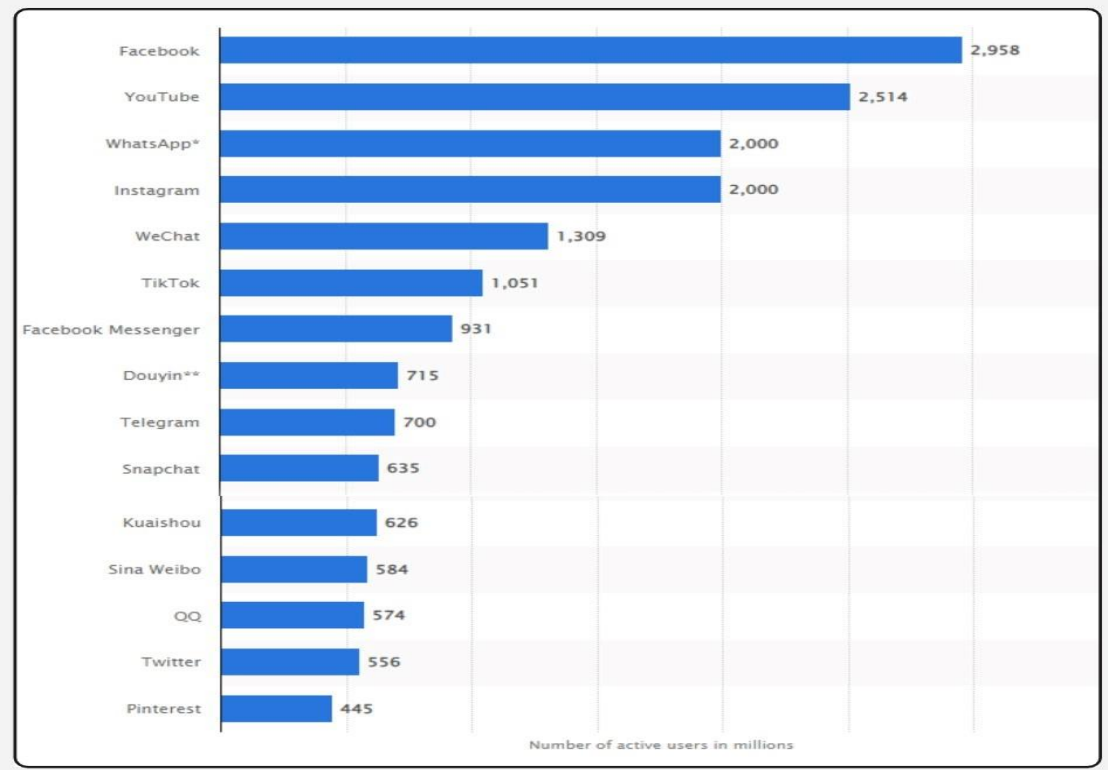

Figure 2.1: The Most Widely Used Social Networks Worldwide as of January 2023, Ranked by the Number of Monthly Active Users.

2.2.1.3 Email advertising

Email advertising is a type of online advertising that involves sending promotional emails to a targeted list of subscribers to promote products or services [20]. This can include newsletters, promotional offers, product updates, and more. It is often used by businesses to nurture leads and drive sales by building relationships with their subscribers. One of the benefits of email advertising is its high level of personalization. Businesses can segment their email lists based on demographics, interests, behaviors, and more, allowing them to tailor their messages to specific audiences. Additionally, email advertising is highly cost-effective, with a low cost per impression and the ability to reach a large audience with a single email [24].

However, it also has its challenges, including the potential for subscribers to mark emails as spam, which can negatively impact the sender's reputation and deliverability [25]. To mitigate this risk, businesses must ensure that their email campaigns comply with anti-spam laws and that they provide value to their subscribers. They must also carefully monitor and optimize their email campaigns to ensure they are delivering the desired results and getting the most value for their advertising spends.

2.2.1.4 E-commerce advertising

E-commerce advertising is a kind of online advertising that focuses on attracting potential customers to individual e-commerce stores or online shopping portals like Amazon or eBay. Increasing online sales and traffic to e-commerce websites are the two main objectives of e-commerce advertising. [10] have identified many forms of e-commerce advertising, such as search, display, and social media adverts. While search ads are displayed on search engine results pages, display ads are visual advertisements that are put on other websites or platforms. Sponsored posts or display adverts on social media sites like Facebook and Instagram are known as social media ads.

One of the key benefits of e-commerce advertising is its ability to reach a highly targeted audience. Advertisers can target users based on a variety of factors, including demographics, interests, behaviors, and more. Additionally, e-commerce advertising can be highly cost-effective, with advertisers only paying for clicks or impressions. However, e-commerce advertising also has its challenges. Competition for online shoppers can be intense, with many e-commerce platforms and stores vying for customers' attention.

Advertisers must ensure that their ads stand out and provide value to potential customers. In order to guarantee that their efforts are yielding the intended outcomes and optimizing their return on investment, they also need to closely monitor and optimize them.

2.2.2 Types of User Behaviour in Advertising

User behavior in advertising can take many forms, and understanding these different types of behavior is essential for creating effective advertising campaigns. Some of the key types of user behavior that are relevant to advertising include attention, perception, attitude, response, trust, avoidance, cross-cultural differences, demographic differences, emotion, and cognitive processing.

2.2.2.1 Attention and engagement

Attention and engagement are crucial components of user behavior in advertising. Users' ability to focus on an advertisement, also known as attention, is influenced by many factors such as the ad's format, placement, and content [26]. One factor that significantly affects attention is the length of the ad. shorter ads tend to be more attention-grabbing, as users are more likely to watch them in their entirety. For example, research has shown that pre-roll video ads that are 15 seconds or less are more likely to be viewed than ads that are longer. Another important factor that influences attention is the ad's placement. Ads that are placed in areas where users are likely to focus, such as at the top or center of a webpage, tend to receive more attention than those placed in less prominent locations. In addition, the ad's relevance to the user also plays a crucial role in attracting their attention. Users are more likely to pay attention to ads that are relevant to their interests or needs.

Engagement, on the other hand, refers to users' active involvement with an advertisement. Engagement can be measured in different ways, such as the time users spend interacting with an ad, the number of clicks or shares, or the level of emotional response generated by the ad [27]. Factors that influence engagement include the ad's format, message, and tone. For example, interactive ads, such as quizzes or games, tend to generate higher levels of engagement as they encourage users to interact with the ad. Ads that use humor, emotional appeals, or storytelling techniques are also more likely to generate higher levels of engagement, as they create an emotional connection with the user [27]. The ad's message and tone are also important factors that influence engagement, as ads that communicate a clear and concise message and use a relatable tone tend to be more engaging.

2.2.2.2 Perception and attitude

There are many factors that can shape users' attitudes towards ads, including relevance, entertainment value, authenticity, and transparency. For example, users are more likely to have positive attitudes towards ads that are relevant to their interests, and that provide a clear benefit or value proposition.

The context in which an ad is viewed can also have a significant impact on users' attitudes towards it. For example, ads that are shown during a favorite TV show or on a trusted website may be viewed more positively than ads that are shown during a less desirable or unfamiliar context. User-generated content (UGC), such as social media posts or product reviews, can also influence users' attitudes towards ads. If users see positive UGC that aligns with the messaging of an ad, they may be more likely to have a positive attitude towards the ad as well [28].

Ad fatigue, or the feeling of being overwhelmed or annoyed by too many ads, can also have a negative impact on user attitudes towards ads. Users may become less receptive to ads over time if they feel that they are being bombarded with too many messages or if the ads are not relevant or engaging. Finally, targeting and personalization can play a critical role in shaping user attitudes towards ads. Ads that are personalized to the user's interests or preferences may be viewed more positively than ads that are not targeted or relevant. However, there is a fine line between personalization and invasion of privacy, and it's important for advertisers to strike a balance between relevance and user privacy.

2.2.2.3 Trust and credibility

Ad transparency is a critical factor in building user trust and credibility. Users are more likely to trust ads that are transparent about their messaging and intentions, and that provide clear and accurate information about the product or service being advertised [29]. Different ad formats can also impact user perceptions of ad credibility. For example, native ads that blend in with the content of a website or app may be viewed as more trustworthy than banner ads that are clearly separate from the content. Ad placement can also influence user perceptions of ad credibility. Ads that are placed on reputable or well-known websites may be viewed as more credible than ads that are placed on lesser-known or low-quality websites.

Social proof, or the concept that people are influenced by the opinions and actions of others, can also play a role in shaping user perceptions of ad credibility. Ads that include social proof elements, such as customer reviews or testimonials, may be viewed as more credible and trustworthy. Finally, the reputation of the brand behind the ad can also impact user perceptions of ad credibility. Users may be more likely to trust ads from brands that have a strong reputation for quality, reliability, and transparency. On the other hand, ads from unknown or untrustworthy brands may be viewed with skepticism or distrust.

2.2.2.4 Avoidance and ad-blocking

As the use of digital advertising has grown, so too has the practice of ad-blocking. Many users today choose to install ad-blocking software on their devices to avoid intrusive or irrelevant ads [30]. This trend presents a significant challenge to advertisers, who must find ways to engage users without being perceived as intrusive or annoying [31]. In this section, we will explore the factors that influence user ad-blocking behavior, including ad relevance, ad quality, and user control. We will also discuss the impact of ad-blocking on advertising revenue and explore strategies that advertisers can use to overcome these challenges and create more positive user experiences.

Ad-blocking has become increasingly popular in recent years, with many users installing ad-blocking software on their devices to avoid ads altogether [32]. Some of the most common reasons cited for ad-blocking include annoyance, intrusive ads, and concerns about privacy and data security [33]. Ad relevance can play a critical role in determining whether users choose to block ads. Ads that are not relevant to a user's interests or needs may be viewed as intrusive or annoying, while ads that are highly relevant may be more likely to be viewed positively.

Ad quality can also influence user ad-blocking behavior. Ads that are poorly designed, intrusive, or disruptive may be more likely to be blocked by users, while ads that are well-designed, engaging, and non-intrusive may be more likely to be accepted. Giving users control over their ad experiences, such as the ability to skip or close ads, may reduce the likelihood of ad-blocking behavior. This can help to create a more positive user experience and build trust between users and advertisers. Finally, it's important to note that ad-blocking can have a significant impact on advertising revenue, as it reduces the number of impressions and clicks that ads receive. Advertisers may need to adapt their strategies to

account for ad-blocking behaviour, such as by creating more engaging and non-intrusive ads or exploring alternative revenue streams.

2.2.2.5 Click advertising

Click advertising, also known as pay-per-click (PPC) advertising, is a popular form of online advertising that involves displaying ads to users and charging the advertiser based on the number of clicks the ad receives. This type of advertising can be found across various platforms, including search engines, social media, and e-commerce websites. Click ads can be targeted to specific keywords, geographic locations, and demographic groups, making them a highly effective form of targeted advertising. There are various ways in which advertisers use clickable ads to drive traffic, engagement, and conversions.

a. Pay-per-click (PPC) Ads: This type of click advertising involves paying each time a user clicks on an ad. PPC ads are often displayed on search engine results pages and social media platforms [34].
b. Display Ads: are clickable advertisements that appear on websites and applications as images or videos. Display advertisements may be tailored according to a user's surfing preferences or demographic data.
c. Native Ads: Native ads are clickable ads that are designed to match the look and feel of the content they appear alongside. They are often displayed on social media platforms or in sponsored content on websites [34].
d. Video Ads: Clickable video ads are often displayed before or during online video content, such as YouTube videos or streaming services like Hulu.
e. Shopping Ads: Users can click through shopping ads to make direct product purchases from e-commerce platforms like Amazon or Google Shopping.
f. App Install Ads: These ads are clickable and designed to encourage users to download and install mobile apps. They can be displayed on social media platforms, search engines, or within other apps.
g. Rich Media Ads: Rich media ads are clickable ads that incorporate advanced features, such as interactive elements or animations, to engage users.

2.3 MACHINE LEARNING AND ARTIFICIAL INTELLIGENCE

Demis Hassabis, CEO of DeepMind, a Google AI startup, defines AI as the "science of making machines smart". This definition is fitting as AI is an umbrella term for various

applications, including subcategories like ML and deep learning, which have real-world AI applications. ML, a type of AI, uses computer systems to process vast amounts of data, learn and improve without explicit programming. It is the most popular AI type and forms the basis of many AI systems used by marketers. Typically, a set of training data is used to educate a ML system to recognize the correct output for a given input. As more data points are processed, the system improves over time.

2.3.1 Algorithms in ML

In the field of ML and evaluating customer behavior, data is essential. We will go into the significance of algorithms and their function in the ML pipeline in this section. An ML algorithm refers to the approach by which an AI system performs its function, usually by predicting output values based on provided input data [35]. Classification of ML algorithms varies, but typically involves categorizing them based on their intended purpose. This thesis author has adopted the conventional method of classifying algorithms according to the following main categories.

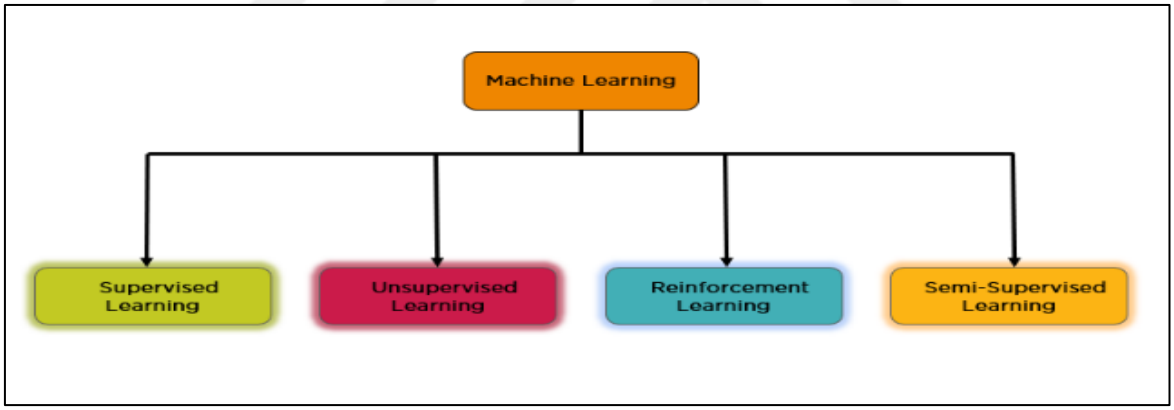

Figure 2.2: ML Algorithmes Types.

2.3.1.1 Supervised learning (SL)

Supervised learning involves feeding the algorithm with training data that contains the desired solutions, referred to as labels [36]. A common example of SL for classification is image recognition. For instance, an algorithm can be trained to identify images of cars and motorcycles by using many example images along with corresponding labels indicating whether they depict a car or a motorcycle. The algorithm then learns how to classify new incoming images.

Regression is another SL task that involves predicting a continuous numerical output [36]. A classic example of regression is predicting the amount of rainfall in a given region based on features such as temperature, humidity, and wind speed. The algorithm is trained on a dataset of previous rainfall data, where each data point contains the features and the corresponding amount of rainfall. The aim is to learn the underlying patterns and relationships between the features and the amount of rainfall, and then use this to predict the amount of rainfall in a new region with similar features.

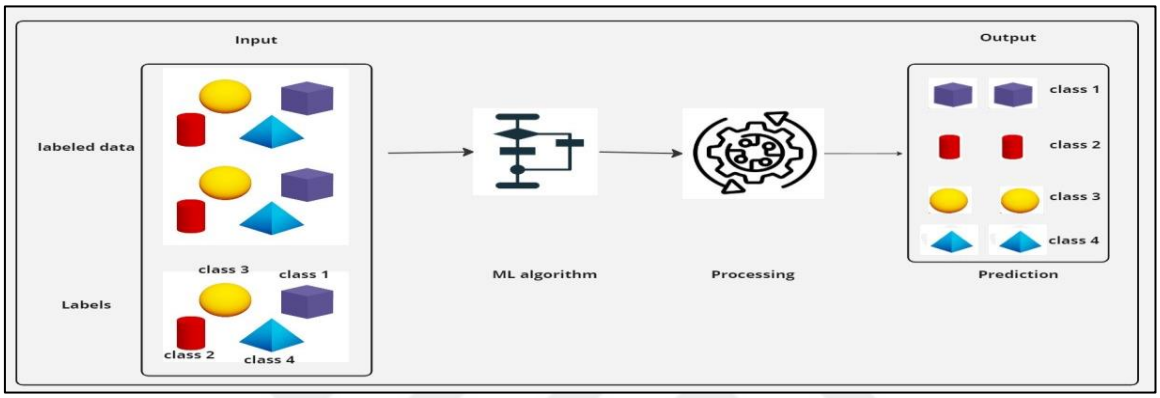

Figure 2.3: Supervised Learning.

2.3.1.2 Unsupervised learning (USL)

In USL (Fig.4), the input data does not have any predefined labels [36]. This is also referred to as unlabeled data, and unlike in SL, the algorithm tries to learn without a teacher [36]. An example of USL is anomaly detection. Imagine a dataset of credit card transactions, some of which may be fraudulent. An anomaly detection algorithm can analyze the patterns in the data and identify the transactions that deviate significantly from the norm, without any prior knowledge of what a fraudulent transaction looks like. This can help financial institutions detect and prevent fraudulent activities.

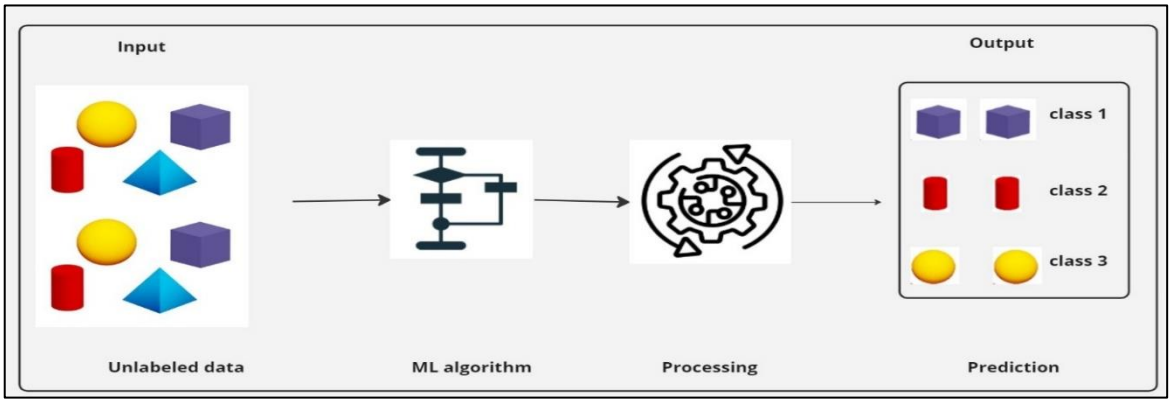

Figure 2.4: Unsupervised Learning.

2.3.1.3 Semi supervised learning (SSL)

SL and USL are two common types of ML algorithms that rely on labeled and unlabeled data, respectively. However, a third class of methods, referred to as SSL, enhances model performance by combining labeled and unlabeled data [36]. A model may be trained on a small collection of labeled data and a much larger set of unlabeled data in the field of natural language processing (NLP), which is one application of SSL. The model uses the small labeled dataset to learn basic language rules, such as grammar and syntax, and then applies these rules to the much larger unlabeled dataset to make predictions about the unlabeled data.

Another example of semi-supervised learning is in anomaly detection, where the goal is to identify rare events or anomalies in a large dataset. In this case, the majority of the data may be unlabeled, but a small subset may be labeled as normal or abnormal [36]. The model can use the labeled data to learn what is considered normal behavior and then apply this knowledge to identify anomalies in the unlabeled data [36]. Google's reCAPTCHA system is another example of SSL, where users are asked to identify objects in images to prove they are human. This labeled data is then used to improve the system's ability to automatically identify objects in images, which is important for preventing automated spam and bots.

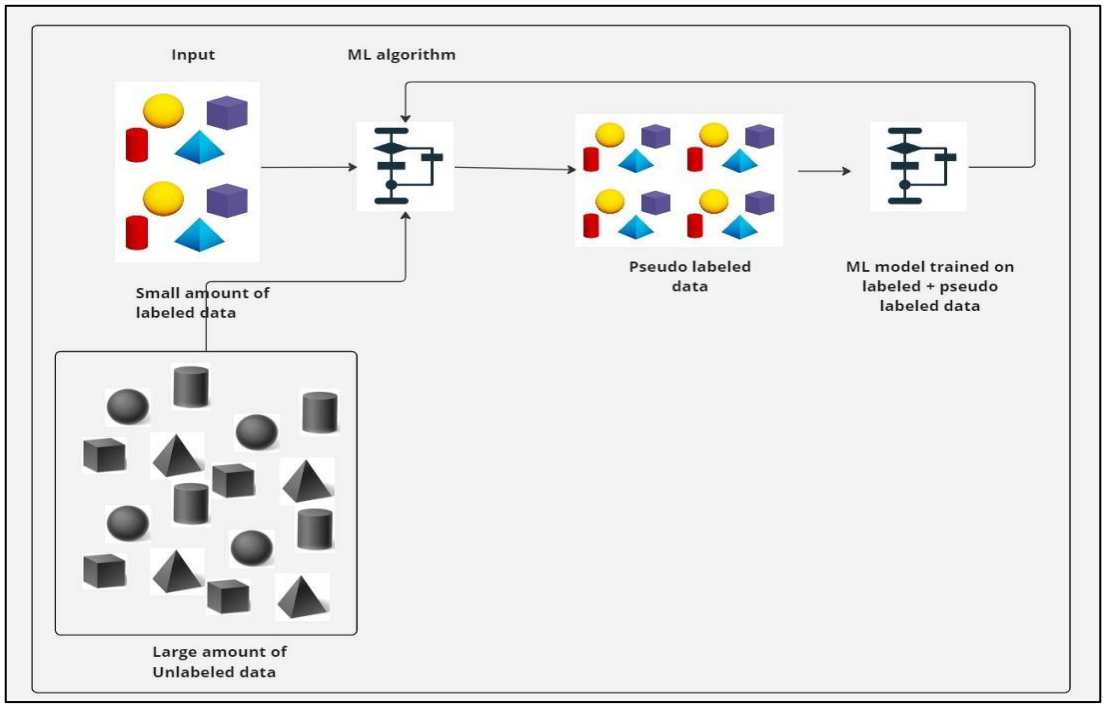

Figure 2.5: Semi-Supervised Learning.

2.3.1.4 Reinforcement learning (RL)

RL is a unique type of ML algorithm that differs significantly from other types. In it, the algorithm learns by interacting with the environment and receiving rewards or penalties for its actions [36]. One example of RL is the development of autonomous vehicles. The algorithm learns how to navigate the environment, avoid obstacles and make decisions based on traffic patterns. Another example is the development of recommendation systems, such as Netflix or YouTube. The algorithm learns the user's preferences by analyzing their viewing history and recommending similar content that the user might like.

2.3.2 Machine Learning Pipeline

An ML pipeline is made up of several linked parts that work together to process data in different ways. These pipelines are frequently employed in ML systems when massive volumes of data need to be altered and converted. [36]. The pipeline components work asynchronously, processing vast amounts of data, and passing it along to other components for further processing, storage, or analysis. This approach makes it easier to maintain and develop the system. If one component fails, it does not affect the entire system, allowing the other components to continue functioning. The ML pipeline typically involves formulating the problem, collecting and evaluating the data, preparing the data for analysis, training and evaluating the model, and finally, visualizing and gaining insights from the results. This pipeline can be customized to fit specific ML projects and can incorporate different algorithms and techniques to achieve the desired outcomes.

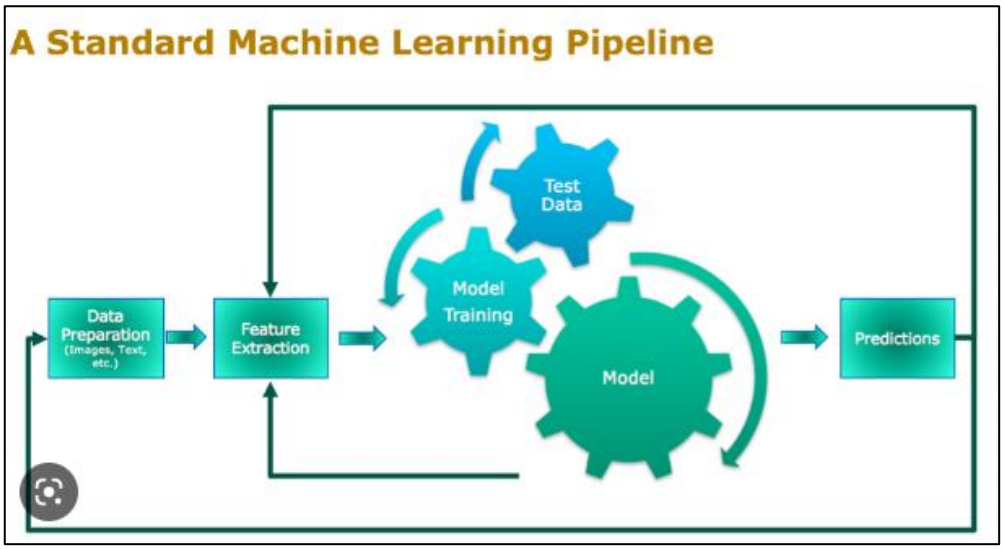

Figure 2.6: ML Pipeline.

2.4 MACHINE LEARNING ALGORITHMS

2.4.1 Ensemble Learning (EL)

A ML approach called EL mixes different models to increase the reliability of predictions. The fundamental idea of EL is that a stronger model, which performs better on unobserved data, may be produced by integrating the predictions of numerous weak models (Dietterich, T. G. 2000). It has become increasingly popular in recent years due to its ability to improve the accuracy and robustness of predictions, as well as its ability to handle complex data and non-linear relationships between input variables and target variables. There are several types of EL methods, including bagging, boosting, voting, and stacking:

a. Bagging: or Bootstrap Aggregating [37], involves training several models independently on different subsets of the training data and then combining their predictions.
b. Boosting: involves iteratively adding new models to the ensemble, each one correcting the errors of the previous ones [38].
c. Voting: is the process of aggregating the forecasts from several models to arrive at a final forecast. When voting, each ensemble model individually forecasts the result of the input data; the aggregate of the individual projections yields the final prediction. Hard voting and soft voting are the two primary categories of voting methods. A simple majority vote of each individual forecast determines the final prediction in a hard vote. The average of the projected probabilities for each individual model is used to determine the final forecast in soft voting.
d. Stacking: or Stacked Generalization [39], involves training multiple models and using their predictions as input to a meta-model that makes the final prediction.

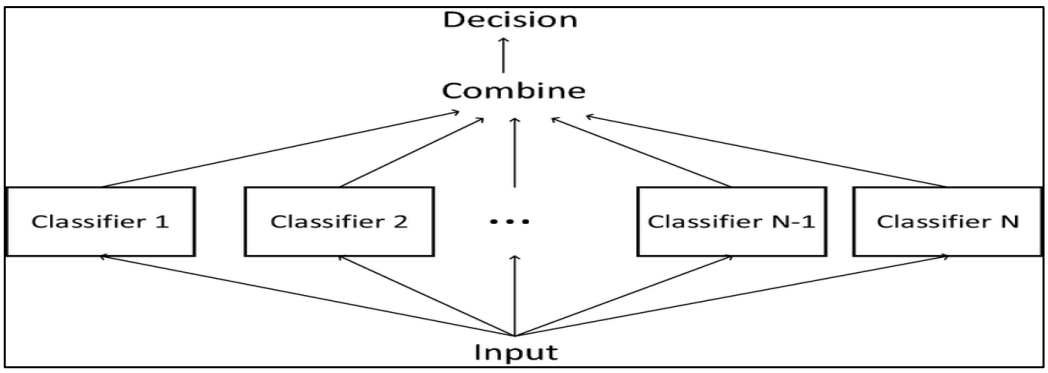

Figure 2.7: Ensemble Learning [40].

2.4.2 Random Forest (RF)

The RF algorithm is an EL method that constructs multiple DTs using randomization [41]. The input is individually classified by each DT, and the final classification is decided by a majority vote cast in each tree. The probabilities at each tree's leaves are averaged to determine the likelihood that an input belongs to a specific class. Unlike traditional DTs, each tree in the forest is grown independently and randomly. The data set used to train each tree is a randomly selected subset of the original training data, with replacement. At each node, a random subset of attributes is used to determine the best split, rather than considering all attributes. The trees in the forest are not pruned and are grown to their full extent. The RF algorithm is efficient and capable of handling large datasets with missing data and outliers.

In the context of the present problem, where a large amount of information is available for each client, using neural networks would require reducing the data to a manageable size by removing potentially significant variables. Instead, the RF algorithm is preferred as it retains the strengths of DTs while overcoming some of their limitations. RF reduces the risk of overfitting as the output of the classifier depends on the entire set of trees rather than a single tree. Randomness is introduced in the creation of each tree, preventing the classifier from memorizing all examples in the training set. Regularization techniques can also be applied to the trees in the forest, further reducing the risk of overfitting. However, like DTs, RF has a bias towards variables with many categories.

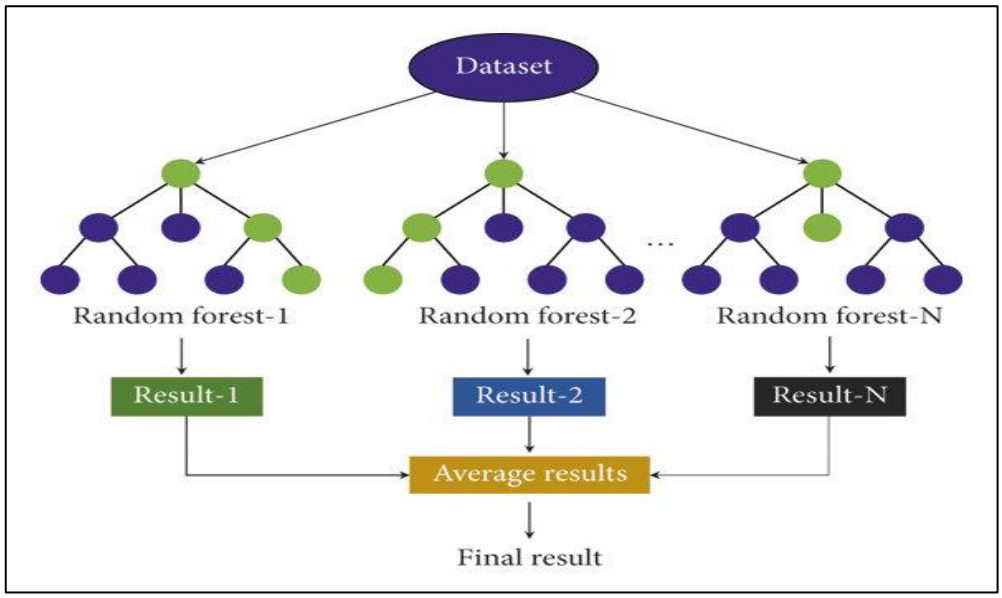

Figure 2.8: Random Forest [42].

2.4.3 Gradient Boosting (GB)

A ML approach called GB is used to create prediction models by fusing a number of weak models [43]. This specific boosting technique functions by continuously adding new models to the ensemble while fixing the shortcomings in the models that came before it. GB's fundamental premise is to train a series of weak models typically DTs on various subsets of the training data. The original data is used to train the first model, while residuals differences between the predicted values and the actual values of the prior models are used to train subsequent models. GB calculates the gradient of the loss function for each iteration in relation to the predictions of the current model. Then, using this gradient, the predictions are updated in a way that minimizes the loss function. GB can build a robust ensemble model that performs well on unobserved data by carrying out this process iteratively. It has the capacity to handle complicated data and non-linear correlations between the input and the target variables, which is one of its benefits. As it may stop adding models when it starts to perceive declining results, it is also less prone to overfitting than other ML methods.

2.4.4 Logistic Regression (LR)

A statistical approach for binary classification issues is LR. The logistic function, which converts a linear combination of input variables into a probability value between 0 and 1, is used to forecast the likelihood of an event occurring based on a collection of input data [44]. Finding the ideal model parameter values that minimize the discrepancy between the projected probabilities and the actual labels of the training data, it is trained on a labeled dataset. A straightforward and understandable model that can handle erratic data and is less prone to overfitting is LR. It can't handle complicated data, though, since there are non-linear correlations between the input variables and the goal variable.

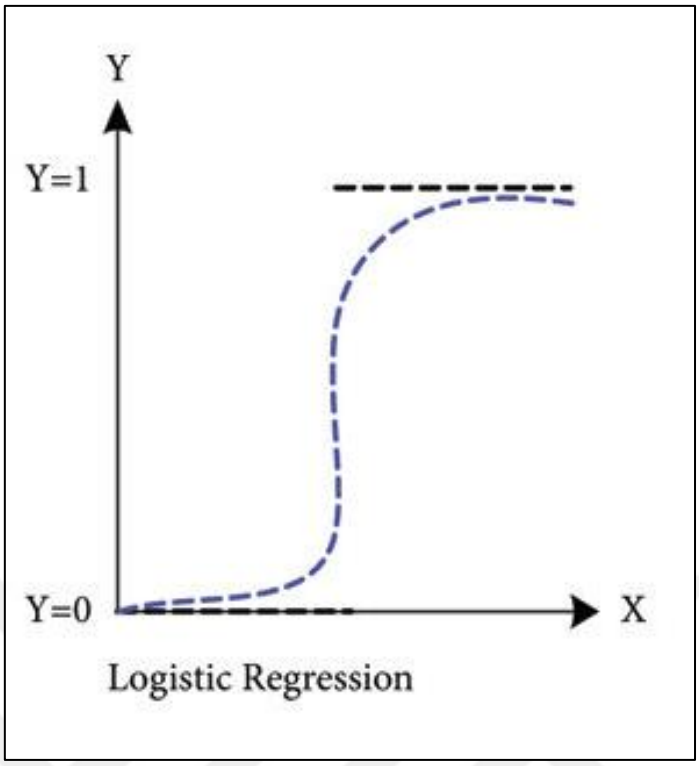

Figure 2.9: Logistic Regression [45].

2.4.5 K-Nearest Neighbors (KNN)

The k-NN model is a simple, intuitive, and non-parametric ML algorithm primarily used for classification, though it can also be applied to regression tasks [46]. Instead of developing an explicit model during the training phase, k-NN operates by storing the entire training dataset in memory. When a prediction is required for an unseen data point, the algorithm searches the training dataset for the "k" training examples that are closest to the point, and returns the most common output value among them for classification or an average for regression. The 'distance' between data points, often calculated using metrics like Euclidean or Manhattan distance, determines the closeness of data points. One of the primary advantages of k-NN is its simplicity and ease of interpretation. However, it can be computationally expensive, especially with large datasets, and is sensitive to irrelevant or redundant features as well as the choice of distance metric. Proper selection of the 'k' value, typically via cross-validation, is crucial for the model's performance.

2.4.6 Decision Tree (DT)

The DT model is a widely-used ML algorithm for both classification and regression tasks. It works by segmenting the data into subsets based on distinctive values of input features

[47]. The tree is constructed in a top-down manner, where the most significant features are placed at the root and branches are created by evaluating the best splits. These splits are determined using metrics like Gini impurity, entropy, or mean squared error, depending on the task. The end nodes, known as leaves, represent the final predictions or decisions. Visually, a DT is easy to interpret and understand, as it mirrors human decision-making processes. One of its primary strengths is its transparent nature, which allows for clear reasoning behind predictions. However, DTs can be prone to overfitting, especially when they are deep, capturing noise and making them less generalizable.

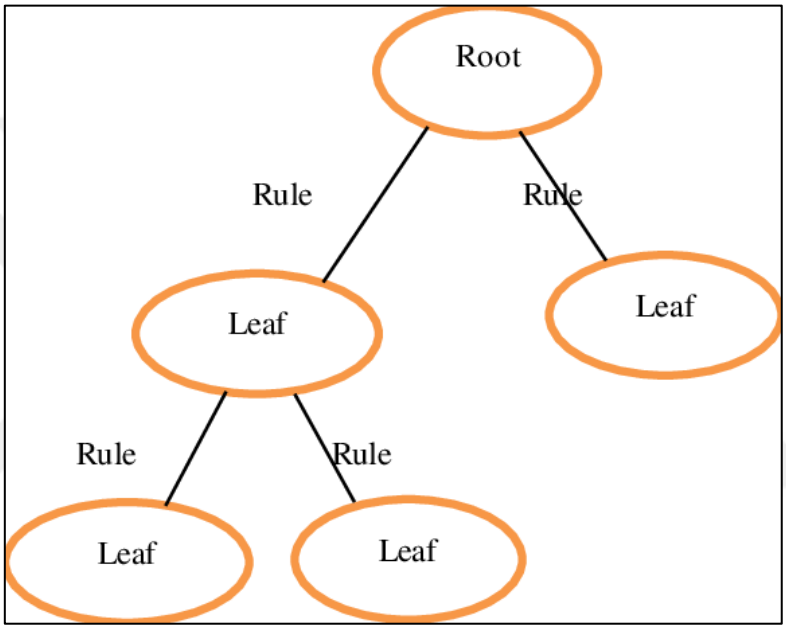

Figure 2.10: Decision Tree [48].

2.4.7 CatBoost

The CatBoost model, which stands for "Category Boosting," is a GB algorithm developed by Yandex, primarily tailored for categorical feature support [49]. Traditional GB models often require extensive preprocessing and encoding of categorical variables, but CatBoost can naturally handle categorical features without the necessity for explicit encoding. Utilizing a technique called "ordered boosting" and a special type of pooling, the model addresses common pitfalls of GB, such as overfitting. Another notable advantage is its efficient and sophisticated handling of missing values. CatBoost also offers built-in support for visualization, which aids in understanding model performance and importance of features. With its robustness against common challenges in handling categorical data and

its capacity to deliver state-of-the-art results, CatBoost has rapidly gained popularity in ML competitions and real-world applications.

2.5 EXPLAINABLE ARTIFICIAL INTELLIGENCE

Explainable Artificial Intelligence, often abbreviated as XAI, represents a paradigm shift in the world of ML and AI [50]. At its core, XAI seeks to make the decisions and workings of AI models transparent, comprehensible, and, as the name suggests, explainable to human users. Traditional AI systems, especially deep learning models like neural networks, have been likened to "black boxes" due to their complex architectures and the obscurity of their internal decision-making processes. While these models are capable of high levels of accuracy, their decisions can be challenging to interpret, which can be problematic in sectors where understanding the rationale behind a decision is crucial, such as healthcare, finance, or criminal justice. XAI aims to address this challenge by creating models that not only predict or classify with high accuracy but also provide clear and intuitive explanations for their decisions. This comprehensibility is crucial for trust, accountability, and validation. If users, be they doctors, analysts, or ordinary individuals, can understand why a model made a particular decision, they can trust its outputs more and can also more effectively diagnose and correct any issues or biases in the system. Furthermore, in many regulated industries, there's a legal or ethical mandate to provide explanations for decisions, making XAI not just a desirable feature but a necessity. As AI and ML continue to permeate various sectors of society, the importance of XAI in ensuring responsible, transparent, and trustworthy AI deployments cannot be overstated.

2.5.1 Explainability and Interpretability

Explainability and interpretability are foundational concepts in the domain of AI and ML that address the transparency and comprehension of algorithmic decision-making [51]. These terms, while sometimes used interchangeably, denote distinct aspects of understanding AI models:

a. Interpretability refers to the extent to which a human can understand the inner workings or mechanisms of a ML model. An interpretable model is one where its decisions can be readily comprehended in human terms and the process it follows to arrive at those decisions is clear. For example, a linear regression model is often deemed interpretable because its decisions are based on weighted input features, and

these weights can be directly inspected and understood. Similarly, DTs lay out a clear, branching path of decisions, making their process transparent. Interpretability is especially crucial in scenarios where stakes are high, like in medical diagnoses or financial predictions, where understanding how a model is making its decisions can be as important as the decisions themselves.

b. Explainability, on the other hand, delves into the realm of post-hoc explanations. It is concerned with how well the internal mechanics of a model can be described to a human, even if those mechanics themselves are not directly interpretable. This is often the case with complex models like deep neural networks, which can act as "black boxes", offering little innate insight into their decision-making processes. Explainability methods aim to provide clarity about these decisions, even if the exact pathway the model took to arrive at them remains obscured. Techniques like Local Interpretable Model-agnostic Explanations (LIME) or SHapley Additive exPlanations (SHAP) are examples of tools that offer explanations for the decisions of complex models, shedding light on the importance of different features or inputs in influencing the model's output.

In essence, while interpretability emphasizes an inherent transparency and direct understanding of a model's operations, explainability focuses on providing understandable reasons for a model's decisions, regardless of the model's internal complexity. Both concepts are pivotal in ensuring that AI and ML models are deployed responsibly, ethically, and effectively, fostering trust and accountability in their applications across various sectors.

2.5.2 Explainable Artificial Intelligence Methods

XAI methods represent a set of techniques and tools designed to elucidate the often opaque decision-making processes of AI and ML models. These methods ensure that the outcomes of these models are transparent, interpretable, and comprehensible to human users, thus fostering trust and accountability. Here's a deep dive into some of the primary methods employed within XAI:

a. LIME: is a technique that approximates black-box classifiers with interpretable models for individual predictions [49]. It operates by perturbing the input data, obtaining predictions for each perturbed instance, and then fitting a locally interpretable model to

predict the black-box model's decisions. The local model's coefficients provide an insight into which features are influencing the prediction.

b. SHAP: inspired by cooperative game theory, attribute the difference between the model's output and its mean prediction to each input feature. It provides a unified measure of feature importance and allocates a value to each feature based on its contribution to a particular prediction.
c. Counterfactual Explanations: These offer insights by answering "what if" questions. For instance, for a loan application denied by an AI model, a counterfactual explanation might state: "The loan would have been approved if the applicant's annual income was $10,000 higher."
d. Feature Visualization: This method involves visualizing high-dimensional data to understand what features an AI model has learned. It's especially useful for neural networks where visualizing the activations and filters can help in understanding what features of the input data the model finds important.
e. Attention Mechanisms in Deep Learning: Originally designed to improve the performance of neural network models, attention mechanisms can also be harnessed for interpretability. They highlight which parts of the input data the model is "focusing" on when making a prediction.
f. Rule Extraction: This involves translating the complex decision boundary of a black-box model into a set of human-readable rules. For example, one could extract DTs or rule lists from neural networks to provide a simpler, interpretable overview of the model's decision process.
g. Model-specific Methods: Some algorithms have built-in methods for interpretability. For instance, DTs are inherently interpretable as the tree structure provides a clear visualization of decision pathways. Similarly, linear regression models offer feature coefficients that indicate the weight or importance of each feature.

3. METHODOLOGY

3.1 PROPOSED METHODOLOGY DESIGN

Our methodology for user behavior detection in online advertising consisted of several key steps (Figure 3.1). We started by obtaining a comprehensive dataset related to online advertising and performed thorough data analysis to gain insights into user behavior. Next, we conducted data preprocessing, which involved cleaning, transforming, and preparing the data for model training. We then implemented base models, including RF, GB, and LR, KNN, DT, and CatBoost both with and without parameter tuning. The models were trained on the preprocessed data and evaluated using metrics. Additionally, we applied ensemble techniques of soft and hard voting to further enhance the models' performance. Soft voting combined the predicted probabilities of the individual models, while hard voting relied on majority voting. Through these steps, we aimed to develop a robust methodology that leveraged ML algorithms and ensemble methods to effectively detect and understand user behavior in online advertising. The evaluation of the models' performance allowed us to identify the best approach, with soft voting exhibiting the highest accuracy, making it the preferred method for our dataset.

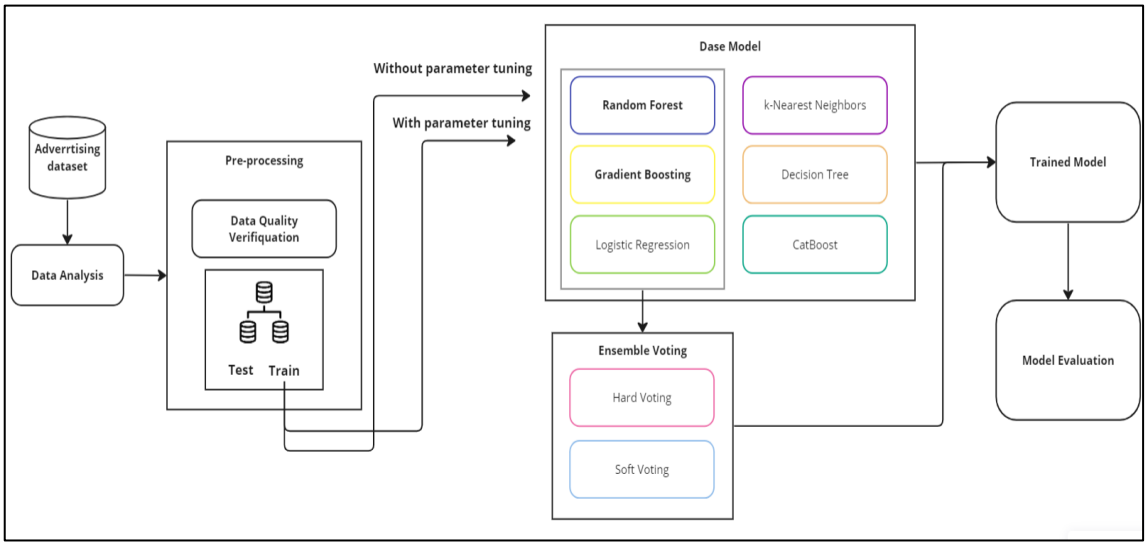

Figure 3.1: The Proposed Methodology Design.

3.1.1 Dataset Gathering

In this study, we utilize a dataset consisting of 1000 instances, each with 8 attributes. Each instance corresponds to a particular user's interaction with an online advertisement. The attributes include:

a. Daily Time Spent on Site: This continuous variable, measured in minutes, denotes the time spent by a user on the site daily.
b. Age: An integer variable representing the age of the user.
c. Area Income: A continuous variable, likely representing the average income in the user's geographic area.
d. Daily Internet Usage: Another continuous variable representing the total time a user spends on the internet daily, possibly measured in minutes.
e. Male: A binary variable indicating the user's gender, where 1 corresponds to male and 0 corresponds to female.
f. The label, or target variable, is "Clicked on Ad", a binary variable where '1' denotes that the user clicked on the advertisement, and '0' indicates that the user did not.

3.1.2 Data Analysis

3.1.2.1 Age distribution

The distribution of ages among our users is a critical component of understanding user interactions with online advertising. Figure 3.2 below shows a histogram illustrating this distribution.

The age of users in our dataset varies from a minimum of 19 years to a maximum of 61 years. The distribution's range helps us to understand the wide diversity in the age of users interacting with the advertisements.

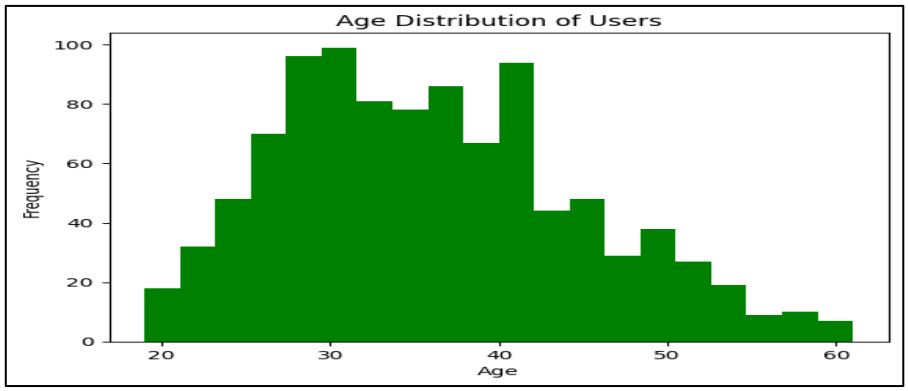

Figure 3.2: Users Age Distribution Histogram.

As we can see from the histogram, the data is bimodal. Most users interacting with the advertisement fall into the 26-32 and 38-40 age brackets. The frequency gradually increases as the age increase from 19-32, indicating a positive correlation between the age of users and their interaction with online advertisements. However, the frequency gradually decreases as the age increase from 40-61, indicating a negative correlation between the age of users and their interaction with online advertisements.

3.1.2.2 Gender distribution

The distribution of gender among our users is another critical component of understanding the demographics interacting with online advertising. Figure 3.3 illustrates a barplot representing the gender distribution among the users in our dataset.

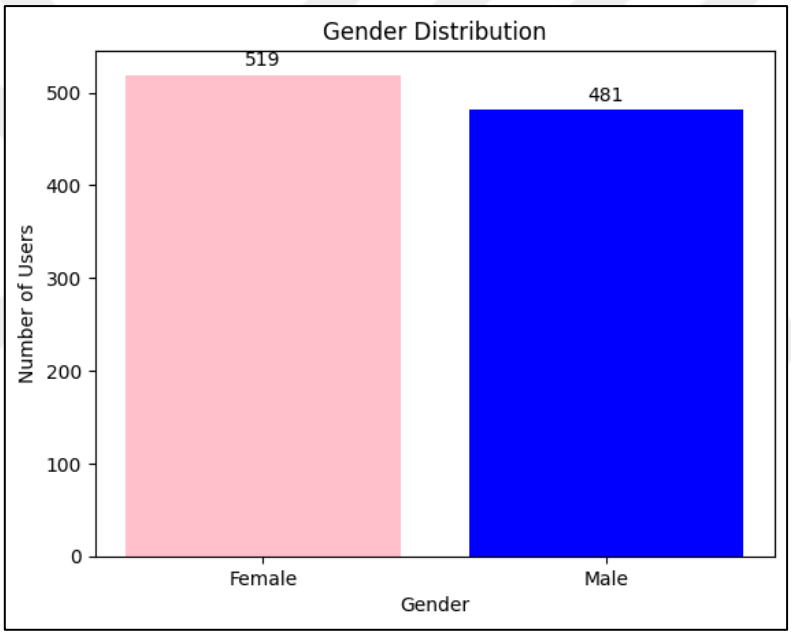

Figure 3.3: Users Gender Distribution.

In our dataset of 1000 users, 519 users, or approximately 51.9% of the sample, are female. The remaining 481 users, or 48.1% of the sample, are male. This relatively balanced gender distribution suggests that our dataset does not significantly bias towards one gender over the other. This allows for a fairer and more reliable analysis when examining gender-related patterns and influences on user interactions with online advertisements.

3.1.2.3 Ad effectiveness

Another key aspect of our study is to understand the effectiveness of the advertisements presented to the users. This is represented by the attribute "Clicked on Ad" where '1'

indicates the user clicked on the advertisement, and '0' indicates they did not. Figure 3.4 illustrates a pie chart displaying the proportion of users who clicked versus those who did not.

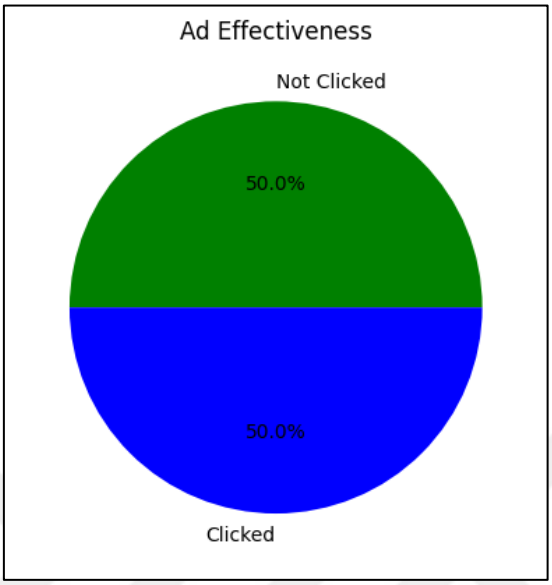

Figure 3.4: Ad Effectiveness Label Distribution.

In our dataset, we observe an equal distribution between users who clicked on the advertisements and those who did not, with both groups comprising 50% of the total users. This balanced distribution is advantageous for our study as it avoids potential bias towards one class over the other in our subsequent predictive modeling.

3.1.2.4 Correlation analysis

To understand the relationships between different attributes in our dataset, we compute a correlation matrix, which is presented in Figure 3.5.

The correlation matrix reveals several key relationships:

a. Daily Time Spent on Site and Clicked on Ad: This pair shows a significant negative correlation of -0.748. This suggests that as the daily time spent on the site increases, the likelihood of the user clicking on an ad decreases. This might imply that users spending more time on the site are perhaps more focused on content rather than the advertisements.
b. Age and Clicked on Ad: There's a positive correlation of 0.493, indicating that as age increases, the likelihood of clicking on an ad also increases. This could suggest that older users are more likely to interact with the advertisements.

c. Area Income and Clicked on Ad: There's a negative correlation of -0.476. This suggests that users from areas with higher incomes are less likely to click on an ad. This could be due to a variety of factors that could be further explored.

d. Daily Internet Usage and Clicked on Ad: This pair shows a strong negative correlation of -0.787. This indicates that users with higher daily internet usage are less likely to click on an advertisement. Like the 'Daily Time Spent on Site' feature, this could suggest that more frequent internet users are less likely to engage with ads.

e. Male and Clicked on Ad: The correlation is -0.038, indicating almost no linear relationship between the user's gender and clicking on an ad.

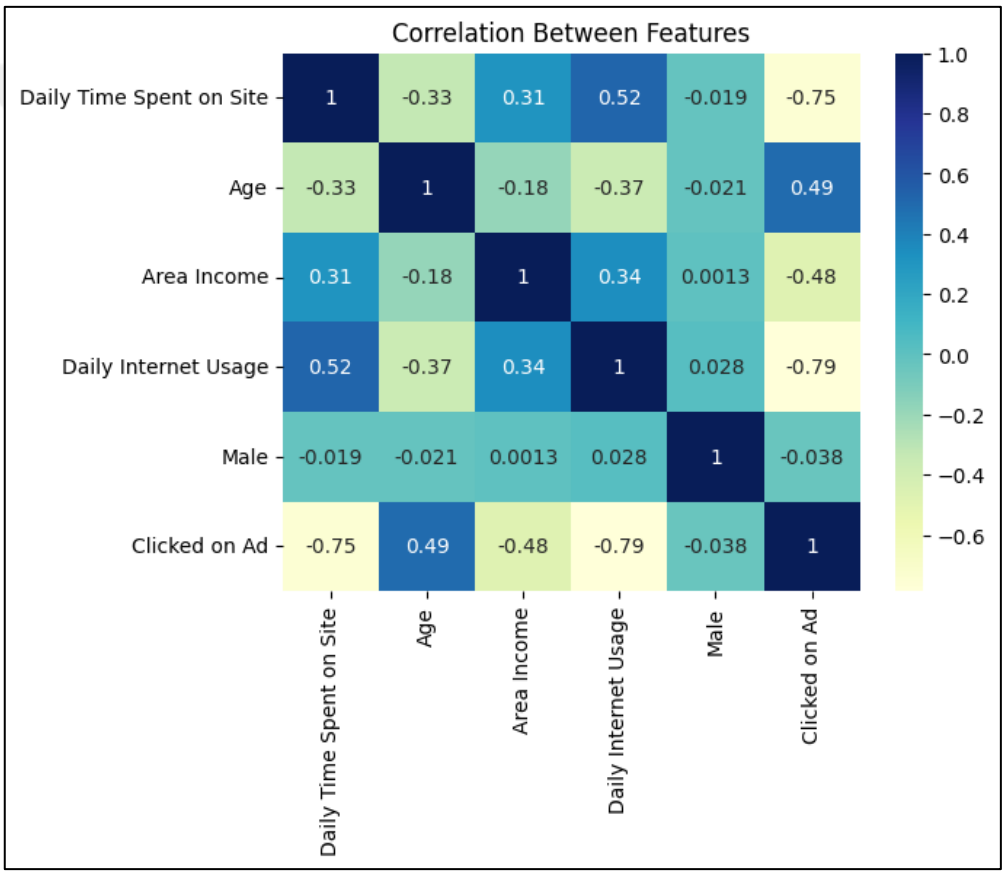

Figure 3.5: Dataset Correlation Matrix.

3.1.3 Data Pre-Processing

3.1.3.1 Data quality verification

Ensuring the quality and integrity of our dataset is a crucial initial step in our research process. As part of this, we verified that there are no missing values or duplicated entries in our dataset.

Specifically, we conducted a thorough data cleaning process by using Python and the pandas library to check for missing values. This was achieved using the *isna().sum()* function on our DataFrame, which indicated that there are no missing (NaN) values across all columns in our dataset.

In addition, we performed a duplicate check on our DataFrame using the duplicated().sum() function. This process confirmed that there are no duplicate entries in our dataset, ensuring the uniqueness of each user interaction record.

This verification of no missing values and no duplicate entries provides a strong foundation for our subsequent data analysis and predictive modeling, guaranteeing that our results and findings are based on accurate and complete data. This step also ensures that our study avoids potential issues related to data quality, such as biased or misleading results due to missing or duplicate data.

3.1.3.2 Data splitting

After ensuring data quality, the next step is data preprocessing and splitting, which prepares our dataset for the subsequent predictive modeling.

This step of data splitting is crucial in the process of building a robust and reliable ML model. It helps us ensure that our model not only fits the training data well but also performs well on new, unseen data, thus minimizing the risk of overfitting.

In our study, we define our feature matrix (X) and the target variable (y). The feature matrix X includes 'Daily Time Spent on Site', 'Age', 'Area Income', 'Daily Internet Usage', and 'Male'. Our target variable y is 'Clicked on Ad', indicating whether a user clicked on an advertisement or not.

The dataset is then split into a training set and a test set using the train_test_split() function from the scikit-learn library. This function shuffles the dataset and then splits it. We allocate 70% of the data to the training set and the remaining 30% to the test set. This split allows us to train our model on a large portion of the data, while still leaving a substantial amount of data unseen for testing and evaluation purposes.

3.1.4 Proposed Classifiers

3.1.4.1 Random forest classifier

The RF classifier was one of the ML models utilized in this research. RF is an EL method which operates by constructing a multitude of DTs at training time and outputting the class that is the mode of the classes (classification) of the individual trees.

Hyperparameters are settings that can be tuned to control the behavior of a ML algorithm. The process of hyper-parameter tuning entails finding the combination of hyperparameters for a learning algorithm that performs the best for a specific problem.

In the case of our RF classifier, five key hyperparameters were selected for tuning:

a. “n_estimators”: determines the number of trees in the forest
b. “max_depth”: controls the maximum depth of each tree.
c. "min_samples_split": This indicates the bare minimum of samples needed to split an internal node when building a tree. Higher values can stop overfitting by making a split require more samples.
d. "min_samples_leaf": This indicates the bare minimum of samples needed at a leaf node. Higher values can assist regulate the complexity of the tree and prevent overfitting, much like'min_samples_split'.
e. "max_features": This variable establishes how many features each node's optimal split should take into account. You can choose between "auto," "sqrt," "log2," or an integer number. Reducing the association between trees and enhancing the diversity of the forest can be achieved by designating a smaller number or a particular feature selection approach.

To find the best hyperparameters, we employed the technique of Grid Search. It is a hyperparameter tuning method that involves exhaustively searching through a specified subset of hyperparameters. We defined a set of potential values for each parameter and the Grid Search method evaluated the RF model performance for each possible combination of these hyperparameters.

```
param_grid = {
 'n_estimators': [50, 100, 200],# Number of trees in the forest
 'max_depth': [None, 10, 20, 30],# Maximum depth of the tree
 'min_samples_split': [2, 5, 10],# Minimum number of samples required to split an internal node
 'min_samples_leaf': [1, 2, 4],  # Minimum number of samples required to be at a leaf node
'max_features': ['auto', 'sqrt', 'log2'],  # Number of features to consider when looking for the best split
}
```

Figure 3.6: Random Forest Parameters Grid.

The best hyperparameters for the RF classifier were found to be {'max_depth': 20, 'max_features': 'log2', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 50}, and the model was subsequently trained using these parameters for the final ensemble model.

3.1.4.2 Gradient boosting classifier

In our investigation, we also made use of the GB classifier, which is an ML model. A group of weak prediction models, usually DTs, are combined to construct a prediction model using the GB EL technique. By permitting the optimization of any differentiable loss function, it expands on the model's step-by-step construction.

In this context, we focused on tuning six key hyperparameters: n_estimators and learning_rate.

a. n_estimators is a representation of the total number of boosting steps needed. A little number could result in underfitting, while a large number could cause overfitting.
b. Each tree's contribution to the ensemble is regulated by learning_rate. To model every relation, a lower learning rate needs more trees, but the predictions are frequently more accurate.
c. "max_depth": This variable indicates the maximum depth that each distinct regression estimator is permitted to have. While increasing'max_depth' may result in a more complicated model, it also raises the possibility of overfitting. Considering the intricacy of the issue and the information at hand, think about testing out several numbers.
d. "min_samples_split": When constructing a tree, this variable determines the absolute minimum number of samples required to split an internal node. Higher values prevent overfitting by increasing the number of samples needed for a split.
e. "min_samples_leaf": This indicates the bare minimum of samples needed at a leaf node. Higher values can aid in preventing overfitting and controlling the complexity of the separate regression estimators.
f. "subsample": This denotes the percentage of samples that will be utilized to fit each unique tree. A portion of the training data is used to incorporate stochasticity into the model for values less than 1.0. This can improve the model's generalization to new data and lessen overfitting.

We implemented Grid Search to optimize these hyperparameters. Grid Search involves exhaustively considering all parameter combinations and retaining the best combination that gives the highest performance, based on a specified scoring metric.

```
param_grid = {
    'n_estimators': [50, 100, 200],  # Number of boosting stages (trees) to perform
    'learning_rate': [0.01, 0.1, 1],  # Learning rate shrinks the contribution of each tree
    'max_depth': [3, 5, 7],  # Maximum depth of the individual regression estimators
    'min_samples_split': [2, 4, 6],  # Minimum number of samples required to split an internal node
    'min_samples_leaf': [1, 2, 3],  # Minimum number of samples required to be at a leaf node
    'subsample': [0.6, 0.8, 1.0],  # Fraction of samples used for fitting the individual trees
}
```

Figure 3.7: Gradient Boosting Parameters Grid.

The optimal hyperparameters for the GB classifier were determined to be { `'learning_rate': 1, 'max_depth': 3, 'min_samples_leaf': 2, 'min_samples_split': 6, 'n_estimators': 100, 'subsample': 0.8`}. These parameters were then used to train the model for the final ensemble.

3.1.4.3 Logistic regression classifier

LR is a popular SL algorithm used for binary classification problems. It is a linear model that uses a logistic function to model the relationship between the input features and the probability of belonging to a specific class. In LR, the input features are combined linearly using weights, and then passed through a logistic (sigmoid) function that maps the linear combination to a probability between 0 and 1. The probability denotes the chance of the sample falling into the positive category. The sample is categorized as belonging to the positive class (clicked) if the probability is higher than a predetermined threshold, which is usually 0.5; if not, it is categorized as belonging to the negative class (non-clicked). Here, we concentrated on adjusting the following five crucial hyperparameters:

a. "C" and "penalty." 'C' represents the inverse of the regularization strength. It controls the trade-off between fitting the training data well and avoiding overfitting. Smaller values of 'C' indicate stronger regularization, while larger values indicate weaker regularization.
b. 'penalty' specifies the type of regularization used in the LR model. 'l1' refers to L1 regularization, also known as Lasso regularization, which encourages sparsity by

shrinking some coefficients to exactly zero. 'l2' refers to L2 regularization, also known as Ridge regularization, which shrinks the coefficients towards zero without making them exactly zero.

c. 'fit_intercept': It determines whether to include an intercept term in the LR model. If set to True, the model will learn an intercept term, and if set to False, no intercept will be included. The choice of including an intercept depends on the problem and the data at hand.
d. 'solver': It specifies the algorithm to use for optimization. Different solvers are available in scikit-learn for LR, each with its own characteristics and strengths. 'liblinear' and 'saga' are two popular solvers, with 'liblinear' being suitable for small-to-medium-sized datasets and 'saga' being suitable for large datasets.
e. 'max_iter': It denotes the maximum number of iterations for the solver to converge. Convergence occurs when the optimization algorithm reaches a stable solution. If the solver does not converge within the specified maximum iterations, the model may not have reached an optimal solution. You can adjust this parameter based on the convergence behavior of the model.

By trying out different combinations of the different parameters from the parameter grid, one can identify the combination that results in the best model performance. This process helps in finding the hyperparameter values that generalize well to unseen data and can improve the performance of this model.

We implemented Grid Search to optimize these hyperparameters.

```
param_grid = { 'C': [0.001, 0.01, 0.1, 1, 10],  # Inverse of regularization strength
'penalty': ['l1', 'l2'],  # Regularization type
'fit_intercept': [True, False],  # Whether to include an intercept term in the model
'solver': ['liblinear', 'saga'],  # Algorithm to use for optimization
'max_iter': [100, 200, 500],  # Maximum number of iterations for the solver# Add more parameters here as
needed
}
```

Figure 3.8: Logistic Regression Parameters Grid.

The optimal hyperparameters for the GB classifier were determined to be: `{ 'C': 1, 'fit_intercept': True, 'max_iter': 100, 'penalty': 'l1', 'solver': 'liblinear'}`. These parameters were then used to train the model for the final ensemble.

3.1.4.4 K-nearest neighbors classifier

the KNN classifier, a non-parametric and instance-based learning algorithm, was explored. The primary mechanism of KNN involves identifying the "k" closest data points (or neighbors) from the training set for each test point and assigning the most frequent label among these neighbors as the predicted class.

To ensure the most optimized performance of the k-NN classifier, a systematic hyperparameter tuning was carried out.

The parameters under consideration included:

a. n_neighbors: The number of neighbors to be taken into account, with tested values being 1, 3, 5, and 7.
b. weights: The strategy for assigning weights to the neighbors, evaluated under two schemes - 'uniform' (where all neighbors are given equal weight) and 'distance' (where closer neighbors are given more weight).
c. p: The power parameter for the Minkowski distance metric, determining the type of distance calculation. A value of 1 corresponds to the Manhattan distance, while a value of 2 represents the Euclidean distance.

To comprehensively assess the various combinations of these hyperparameters and determine the best configuration, the GridSearchCV method was employed. This method performs exhaustive testing over the specified parameter values, ensuring that the final selected hyperparameters are those that provide the highest performance on the training data, under a 3-fold cross-validation setup.

```
# Define the parameter grid for KNN
param_grid = {
    'n_neighbors': [1, 3, 5, 7],  # Number of neighbors to consider
    'weights': ['uniform', 'distance'],  # Weighting scheme for neighbors
    'p': [1, 2]  # Power parameter for the Minkowski distance metric (1 for Manhattan, 2 for Euclidean)
}
```

Figure 3.9: KNN Parameters Grid.

Upon the completion of this grid search, the most optimal settings for the k-NN classifier were found to be {'n_neighbors': 5, 'p': 1, 'weights': 'distance'}, indicating that considering 5 neighbors, using Manhattan distance (p=1), and assigning weights based on distance achieved the best results for the problem addressed in the thesis.

3.1.4.5 Decision tree classifier

The DT is a flowchart-like structure where internal nodes represent features, branches denote decisions, and leaf nodes signify outcomes or class labels. The algorithm makes decisions based on asking a series of questions and following the path that aligns with the answer, leading to a classification outcome.

To maximize the performance of the DT classifier and tailor it to the specifics of the dataset and problem, a detailed hyperparameter tuning process was undertaken. The hyperparameters considered for optimization included:

a. criterion: The function used to measure the quality of a split. The study evaluated both 'gini' for Gini impurity and 'entropy' for information gain.
b. max_depth: The maximum depth of the tree, with potential values being None (indicating no limit), 10, 20, and 30.
c. min_samples_split: Specifies the minimum number of samples required to split an internal node, tested for values 2, 5, and 10.
d. min_samples_leaf: Determines the least number of samples necessary for a leaf node, with candidate values being 1, 2, and 4.

To find the optimal combination of these hyperparameters, the GridSearchCV method was employed. This method conducts an exhaustive search over the specified hyperparameters to determine which combination offers the best performance, validated using a 3-fold cross-validation strategy. After fitting the DT model with different parameter combinations on the training data, GridSearchCV ensures that the selected hyperparameters deliver the best predictive performance for the given problem.

```
param_grid = {
    'criterion': ['gini', 'entropy'],  # Splitting criterion
    'max_depth': [None, 10, 20, 30],   # Maximum depth of the tree
    'min_samples_split': [2, 5, 10],   # Minimum samples required to split an internal node
    'min_samples_leaf': [1, 2, 4]      # Minimum samples required at a leaf node
}
```

Figure 3.10: DT Parameters Grid.

The optimal hyperparameters for the DT classifier were determined to be:
{'criterion': 'gini', 'max_depth': 30, 'min_samples_leaf': 2, 'min_samples_split': 10}

3.1.4.6 CatBoost classifier

In the context of this thesis, the CatBoost classifier, an advanced GB algorithm, was explored. Originating from Yandex, CatBoost stands out primarily due to its built-in support for categorical features, eliminating the need for extensive preprocessing or manual encoding which is required in many other ML algorithms.
For the purpose of fine-tuning the CatBoost classifier to best fit the study's data and problem specifics, a hyperparameter optimization was undertaken.

The considered hyperparameters encompassed:

a. iterations: Refers to the number of boosting stages the algorithm should run, with values explored being 100, 200, and 300.
b. learning_rate: Dictates the speed of the model's training, with candidate values being 0.01, 0.1, and 0.2.
c. depth: Represents the depth of the trees in the model, and the study evaluated depths of 4, 6, and 8.
d. l2_leaf_reg: An L2 regularization term on weights, introduced to reduce overfitting, with tested coefficients being 1, 3, and 5.

To ascertain the most optimal combination of these hyperparameters, the GridSearchCV method was employed. This method performs a comprehensive search over the provided hyperparameter values to identify the combination that results in the best performance. Using a 3-fold cross-validation strategy, GridSearchCV validates the efficacy of each combination, ensuring the selected hyperparameters are the most suitable for the task at hand.

```
# Define the parameter grid for CatBoostClassifier
param_grid = {
    'iterations': [100, 200, 300],        # Number of boosting iterations
    'learning_rate': [0.01, 0.1, 0.2],    # Learning rate
    'depth': [4, 6, 8],                   # Depth of the tree
    'l2_leaf_reg': [1, 3, 5],             # L2 regularization coefficient
}
```

Figure 3.11: Catboost Parameters Grid.

The optimal hyperparameters for the CatBoost classifier were determined to be:
(depth= 4, iterations= 300, l2_leaf_reg= 3,learning_rate=0.01)

3.1.4.7 Ensemble model

After applying the base models and fine-tuning their parameters using the data, we further enhanced our analysis by employing ensemble voting techniques, specifically soft and hard voting. The purpose of ensemble voting was to leverage the collective wisdom of the individual models and explore whether these techniques could yield even better results for our data.

Soft voting, which combines the predicted probabilities of the individual models, was applied to generate a final prediction. This method takes into account the confidence level of each model's predictions, allowing for a more nuanced and probabilistic decision-making process. On the other hand, hard voting, which aggregates the class votes of the individual models, relies on the majority vote to make the final prediction. This approach simplifies the decision-making process by considering only the most common prediction.

By applying soft and hard voting to our fine-tuned models, we aimed to assess their effectiveness in improving prediction accuracy and performance. This analysis provides insights into the strengths of each voting method and helps determine the best approach for our specific data.

4. RESULTS AND DISCUSSION

4.1 THE USED LIBRARIES

In our study, we utilized several libraries to facilitate our analysis and implementation of ML models.

The libraries used include:

a. Pandas is a powerful library for data manipulation and analysis. It provides convenient data structures and functions to efficiently work with structured data, such as data frames. We used Pandas to load and preprocess our dataset, perform data exploration, and create training and testing data sets.
b. NumPy is a fundamental library for numerical computations in Python. It provides support for multi-dimensional arrays and various mathematical functions. We leveraged NumPy to perform numerical operations and transformations on the data, such as scaling and array manipulation, to prepare it for model training.
c. Matplotlib is a popular library for data visualization in Python. It offers a wide range of plotting functions and customization options. We utilized Matplotlib to create visualizations, such as line plots, scatter plots, and bar charts, to gain insights into the data and present the results of our analysis.
d. Seaborn is a high-level data visualization library built on top of Matplotlib. It provides a more aesthetically pleasing and convenient interface for creating statistical graphics. We used Seaborn to generate informative and visually appealing visualizations, such as heatmaps and distribution plots, to explore relationships and distributions within the data.
e. Scikit-learn is a widely used ML library in Python. It offers a comprehensive set of tools for various ML tasks, including model selection, preprocessing, and evaluation. We employed the train_test_split function from Scikit-learn to split our dataset into training and testing sets, allowing us to train our models on a portion of the data and evaluate their performance on unseen data.

4.2 EVALUATION METRICS

The evaluation of ML models involves the use of various metrics to assess their performance and effectiveness. In our study, we employed the grid search method with a

cross-validation of 3 to tune the hyperparameters of the models. By conducting grid search with a cross-validation of 3, we ensured that the models' performance was evaluated in a robust and reliable manner. Let TP represent the number of true positive predictions, TN represent the number of true negative predictions, FP represent the number of false positive predictions, and FN represent the number of false negative predictions.

To evaluate the performance of the models, we used several key metrics:
Accuracy (ACC): measures the overall correctness of the predictions made by the models. It calculates the ratio of correctly predicted instances to the total number of instances.

$$ACC = (TP + TN) / (TP + TN + FP + FN)$$

Precision (PRE): quantifies the model's ability to correctly identify positive instances. It measures the ratio of correctly predicted positive instances to the total number of instances predicted as positive.

$$PRE = TP / (TP + FP)$$

The model's recall (sensitivity) indicates how well it can recognize every positive case. The ratio of accurately predicted positive instances to the total number of real positive instances is computed.

$$REC = TP / (TP + FN)$$

F1-score (F1-S): is the harmonic mean of PRE and REC. It provides a balanced measure of the model's performance.

$$F1\text{-}S = 2 * (PRE * REC) / (PRE + REC)$$

4.3 MODELS PERFORMANCE WITHOUT PARAMETER TUNING

4.3.1 Evaluation of the Random Forest Model

The RF model without parameter tuning achieved an ACC of 97.3%, indicating a high level of overall correctness in its predictions. The model's performance can be further analyzed using the CM and the CR. The CM reveals that out of the total samples, 141 were correctly classified as belonging to class 0, while 151 were correctly classified as belonging to class 1. However, there were 5 instances where the model incorrectly predicted samples as class 1 when they actually belonged to class 0, and 3 instances where samples belonging to class 1 were incorrectly predicted as class 0. The CR provides an assessment of PRE, REC, and F1-S for each class.

The CR for the RF model reveals its strong performance for both classes. With a PRE of 0.98 for Class 0 and 0.97 for Class 1, the model accurately identified the majority of instances for each class. Additionally, the REC values of 0.97 for Class 0 and 0.98 for Class 1 demonstrate the model's ability to effectively capture positive instances. The balanced F1-Ss of 0.97 for both classes indicate a good trade-off between PRE and REC.
It achieved a high sensitivity, indicating that it correctly classified 98.1% of the positive instances. On the other hand, it achieved a specificity of 96.6%, indicating that it correctly classified 96.6% of the negative instances.

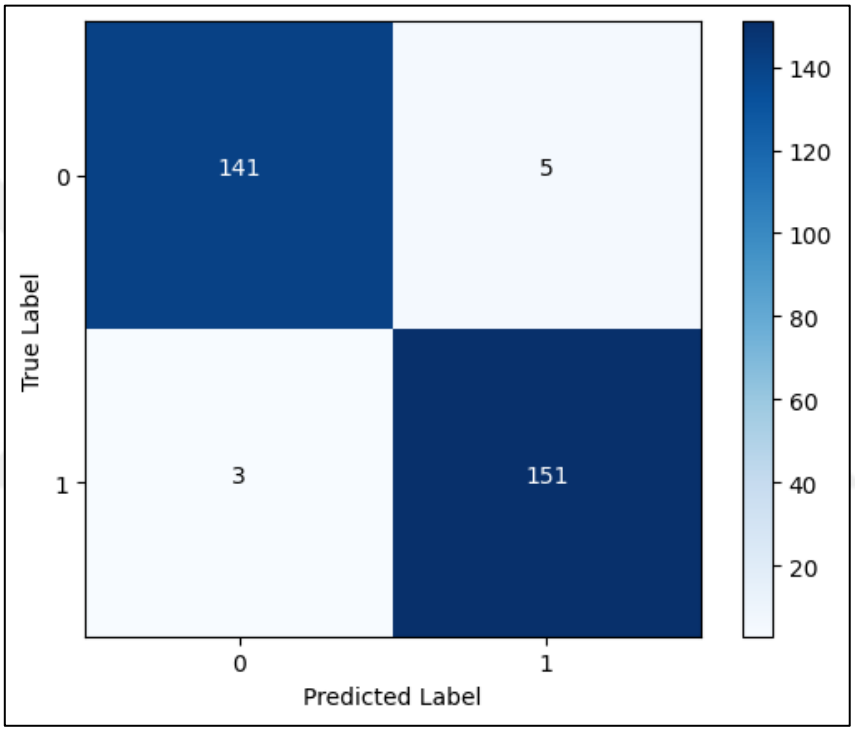

Figure 4.1: Random Forest Confusion Matrix.

Classification Report:

	precision	recall	f1-score	support
0	0.98	0.97	0.97	146
1	0.97	0.98	0.97	154
accuracy			0.97	300
macro avg	0.97	0.97	0.97	300
weighted avg	0.97	0.97	0.97	300

Figure 4.2: Random Forest Classification Report.

Figure 4.3 illustrates the feature importance as determined by the LIME method when applied to a RF model. The figure highlights the significance of various features and their

corresponding weight ranges in influencing the model's decisions. "Daily Internet Usage" within the range of 185.45 to 220.06 has a negative weight, indicating its potential in decreasing the likelihood of the predicted outcome. On the other hand, features like "Age" greater than 41, "Area income" less than or equal to 47,332.82, "Daily Time Spent on Site" between 51.47 and 68.22, and "Male" value less than or equal to 0 showcase positive weights, implying their roles in increasing the probability of the model's prediction. The magnitude of these weights provides insights into the relative importance of each feature in the model's decision-making process.

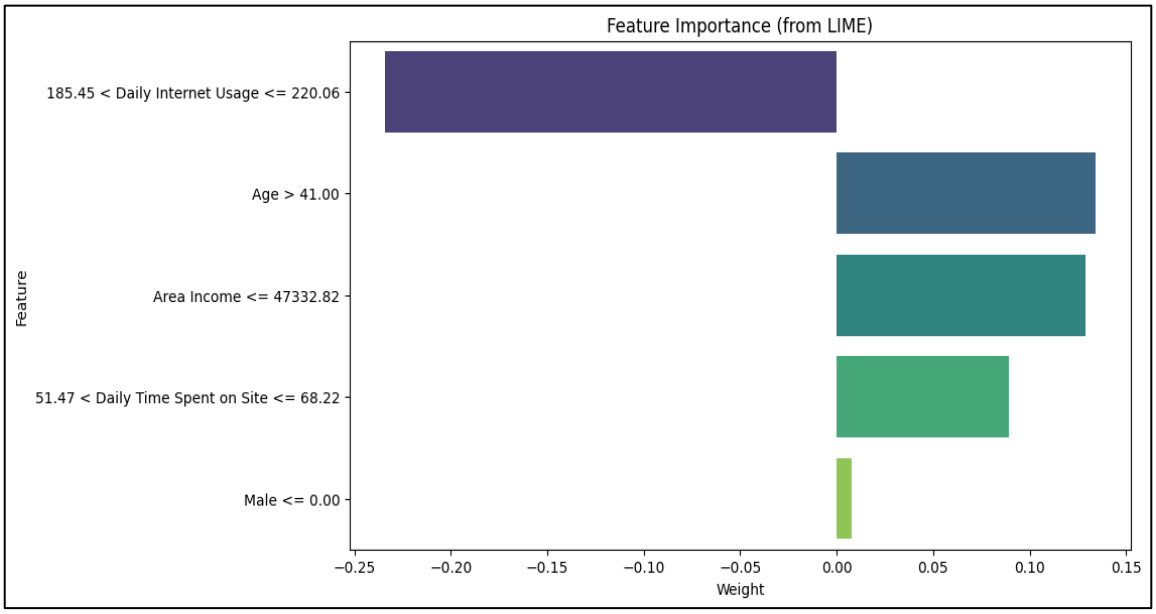

Figure 4.3: Feature Importance Derived from LIME Analysis on a RF Model.

Figure 4.4 displays a vertical bar chart that visualizes the average impact on model output magnitudes derived from SHAP for the RF method, broken down by two distinct classes. Each feature's influence is presented by a pair of bars side by side, with Class 1 represented in blue and Class 0 in red. For "Daily Internet Usage," Class 1 has an influence ranging from 0 to 0.22, followed immediately by Class 0's impact extending from 0.22 to 0.45. The "Daily Time Spent on Site" showcases Class 1's range from 0 to 0.18, and Class 0's influence from 0.18 to 0.38. When observing "Area Income," Class 1's impact spans from 0 to 0.05, with Class 0 following from 0.05 to 0.11. For the "Age" feature, Class 1's influence ranges between 0 to 0.05, slightly overlapping with Class 0's range of 0.04 to 0.1. Lastly, the "Male" feature demonstrates minimal variations, with Class 1's range from 0 to

0.001 and Class 0's from 0.001 to 0.0015. The bar chart succinctly highlights the differences in feature impact for the two classes, providing a visual comparative analysis.

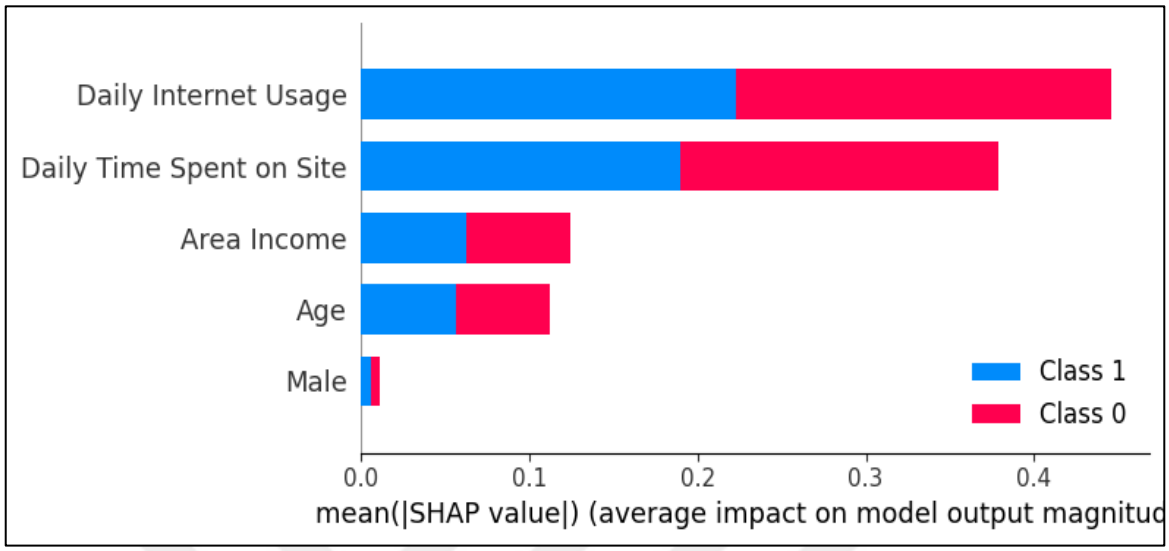

Figure 4.4: Average Impact on Model Output Magnitudes for RF Using SHAP.

4.3.2 Evaluation of the Logistic Regression Model

The LR model without parameter tuning achieved an ACC of 0.967, indicating a high level of overall correctness in its predictions. The CM provides additional insights into the model's performance. Out of the total samples, 142 were correctly classified as belonging to class 0, while 148 were correctly classified as belonging to class 1. However, there were 4 instances where the model incorrectly predicted samples as class 1 when they actually belonged to class 0, and 6 instances where samples belonging to class 1 were incorrectly predicted as class 0.

The CR for the LR model without parameter tuning demonstrates its strong performance for both classes. With a PRE of 0.96 for Class 0 and 0.97 for Class 1, the model accurately identified the majority of instances for each class. Additionally, the REC values of 0.97 for Class 0 and 0.96 for Class 1 highlight the model's ability to effectively capture positive instances. The balanced F1-S of 0.97 for both classes indicate a good balance between PRE and REC.

It achieved a sensitivity of 96.1%, indicating that it correctly classified 96.1% of the positive instances and achieved a specificity of 97.3%, indicating that it correctly classified 97.3% of the negative instances.

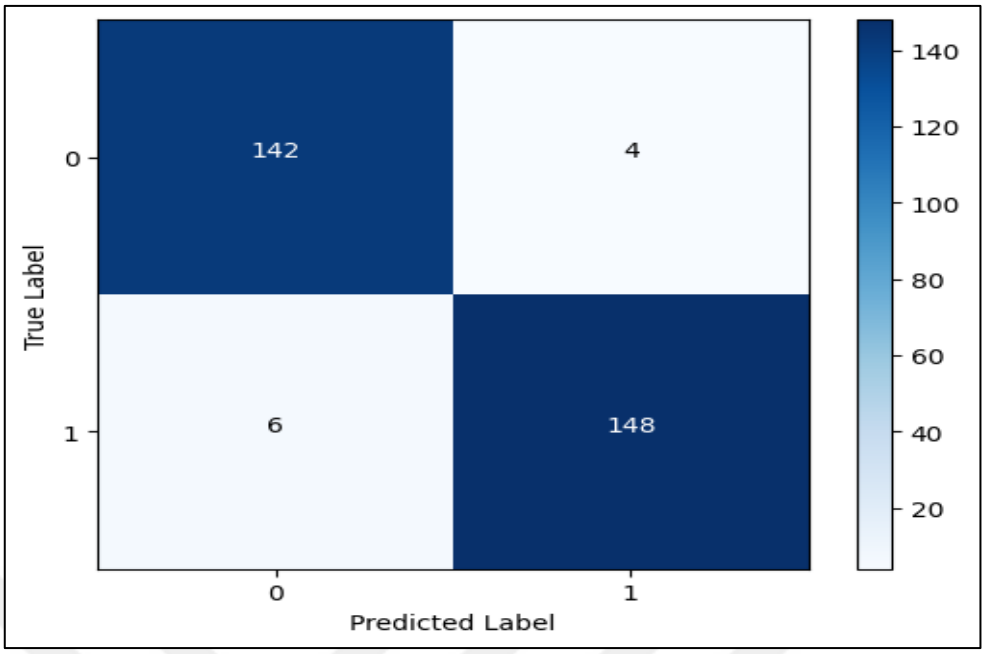

Figure 4.5: Logistic Regression Confusion Matrix.

Classification Report:				
	precision	recall	f1-score	support
0	0.96	0.97	0.97	146
1	0.97	0.96	0.97	154
accuracy			0.97	300
macro avg	0.97	0.97	0.97	300
weighted avg	0.97	0.97	0.97	300

Figure 4.6: Logistic Regression Classification Report.

Figure 4.4 showcases the feature importance as ascertained by the LIME method for a LR model. This visualization delineates the impact of specific features on the model's predictions by highlighting their associated weight ranges. The "Daily Internet Usage" between 185.45 and 220.06, "Age" greater than 41, and "Daily Time Spent on Site" between 51.47 and 68.22 all possess positive weights, suggesting their influence in amplifying the model's predictive outcome. Conversely, "Area income" less than or equal to 47,332.82 has a negative weight, indicating its role in reducing the likelihood of the predicted result. The "Male" feature with a value less than or equal to 0 exhibits a slight negative weight, hinting at its minor contribution in steering the model's decision. This figure provides a concise view of feature significance within the context of a LR model's decision-making process.

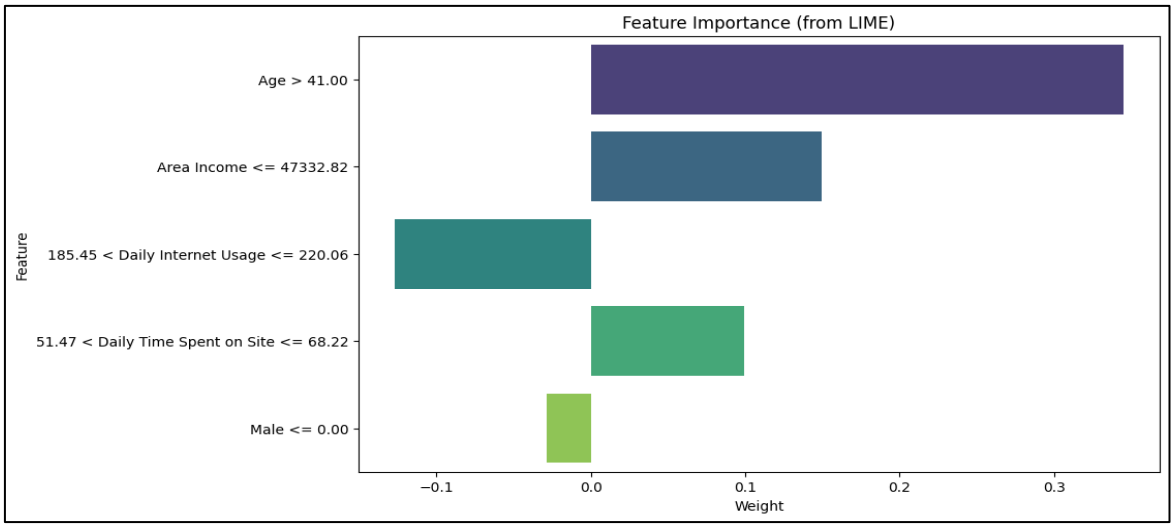

Figure 4.7: Feature Importance Derived from LIME Analysis on a LR Model.

Figure 4.12 illustrates a vertical bar chart that represents the average impact on model output magnitudes derived from SHAP when employing the LR method. In the figure, Class 1 is distinctly represented in blue. The influence of "Daily Internet Usage" for Class 1 stretches from 0 to 0.28. The "Age" feature for Class 1 showcases a notable range, extending from 0 to 2.5. Similarly, the "Daily Time Spent on Site" also holds an impact range of 0 to 2.5 for Class 1. "Area Income" presents a more moderate influence for Class 1, spanning from 0 to 1.0. Lastly, the "Male" feature demonstrates a range of 0 to 0.2 for Class 1. The bar chart offers a visual representation of the features' impacts on the model's output for Class 1, emphasizing their respective magnitudes.

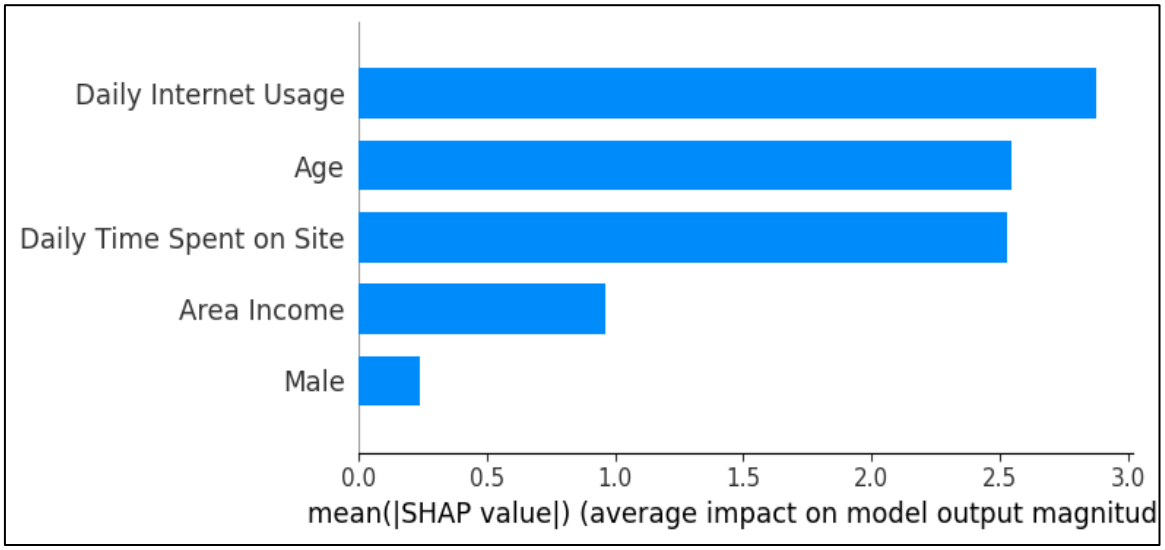

Figure 4.8: Average Impact on Model Output Magnitudes for LR using SHAP.

4.3.3 Evaluation of the Gradient Boosting Model

The ACC achieved by the model is 96.7%. The CM for the GB model without parameter tuning provides insights into its performance. Among the total samples, 138 were accurately classified as class 0, and 152 were correctly classified as class 1. However, there were 8 instances where the model misclassified samples as class 1 when they actually belonged to class 0, and 2 instances where samples from class 1 were mistakenly classified as class 0.

The CR for the GB model without parameter tuning reveals strong performance for both classes. For Class 0, the model achieved a PRE of 0.99, indicating that 99% of the instances predicted as Class 0 were correctly classified. The REC for Class 0 is 0.95, indicating that the model identified 95% of the instances belonging to Class 0 correctly. The F1-score for Class 0 is 0.97, signifying a good balance between correctly identifying Class 0 instances and minimizing false positives. For Class 1, the model achieved a PRE of 0.95, indicating that 95% of the instances predicted as Class 1 were correctly classified. The REC for Class 1 is 0.99, meaning that the model identified 99% of the instances belonging to Class 1 correctly. The F1-score for Class 1 is 0.97, suggesting a good balance between correctly identifying Class 1 instances and minimizing false positives. Results also demonstrate a sensitivity of 0.987 and a specificity of 0.945.

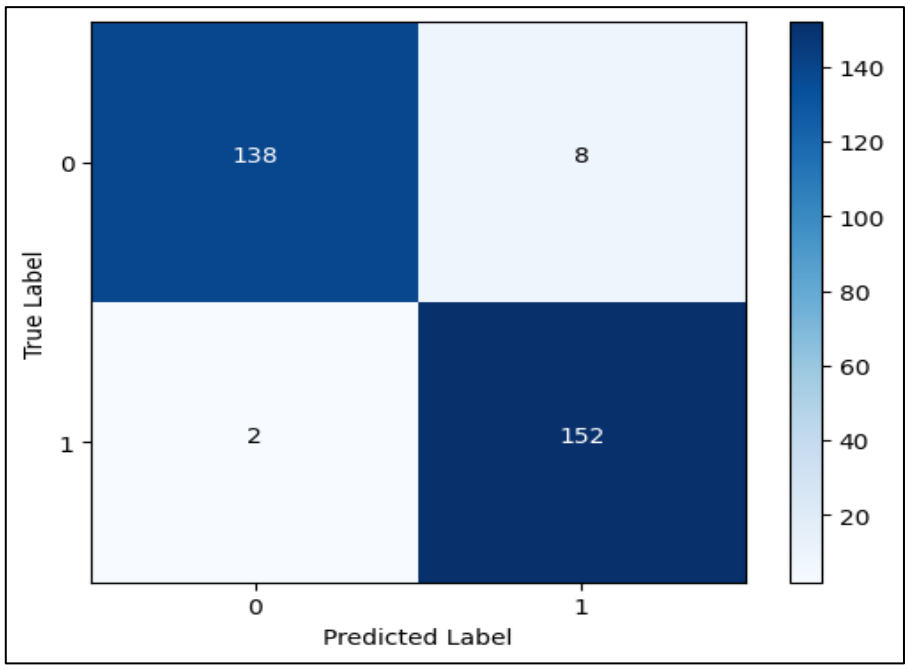

Figure 4.9: Gradient Boosting Confusion Matrix.

Classification Report:

	precision	recall	f1-score	support
0	0.99	0.95	0.97	146
1	0.95	0.99	0.97	154
accuracy			0.97	300
macro avg	0.97	0.97	0.97	300
weighted avg	0.97	0.97	0.97	300

Figure 4.10: Gradient Boosting Classification Report.

Figure 4.11 displays the feature importance as deduced from LIME analysis when applied to a GB model. The figure underscores the influence of distinct features in shaping the model's decisions by indicating their respective weight intervals. The "Daily Internet Usage" between 185.45 and 220.06 holds a negative weight, hinting at its role in diminishing the predicted outcome's likelihood. Conversely, the features "Age" exceeding 41, "Area income" not surpassing 47,332.82, "Daily Time Spent on Site" spanning from 51.47 to 68.22, and "Male" equaling 0 or less, all exhibit positive weights, signifying their contribution to increasing the probability of the model's prediction. The magnitude of these weights provides a snapshot of the relative importance each feature holds within the GB model's decision framework.

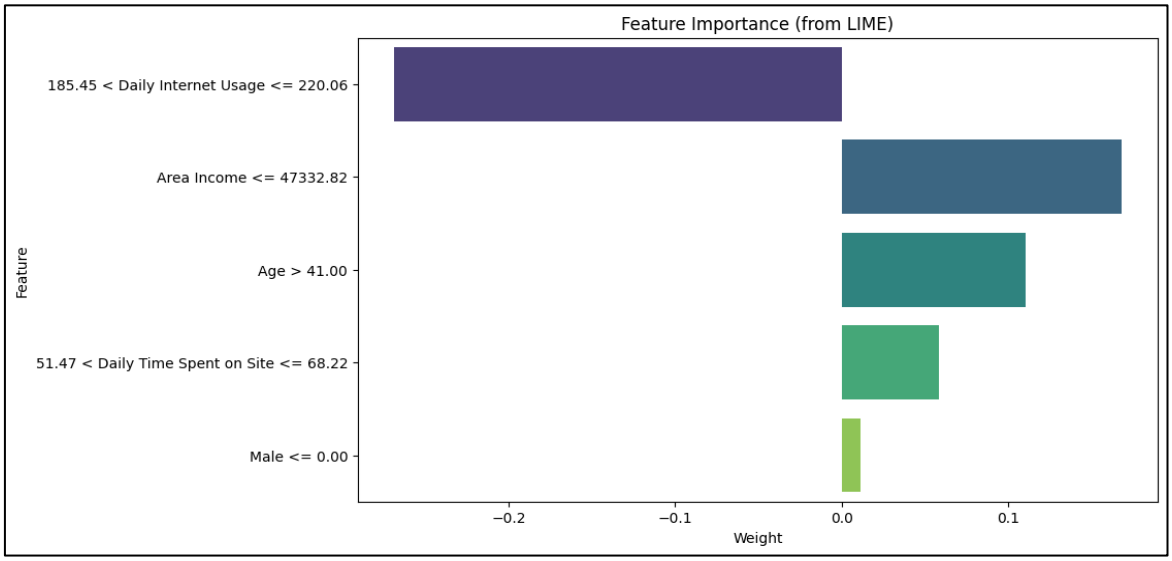

Figure 4.11: Feature Importance Derived from LIME Analysis on a GB Model.

Figure 4.12 displays a vertical bar chart showcasing the average impact on model output magnitudes using the SHAP values for the GB method. In this visual representation, Class 1 is distinctly marked in blue. The "Daily Internet Usage" for Class 1 ranges from 0 to

0.25, indicating its influence on the model's output. For "Daily Time Spent on Site", the impact spans from 0 to 2.0 for Class 1. "Area Income" demonstrates an influence ranging from 0 to 0.8 for Class 1. The "Age" feature, meanwhile, has an impact stretching from 0 to 0.5 for Class 1. Lastly, the "Male" attribute holds a more contained range, from 0 to 0.1, for Class 1. The chart succinctly captures the significance of each feature's impact on the model's predictions for Class 1 when applying the GB technique.

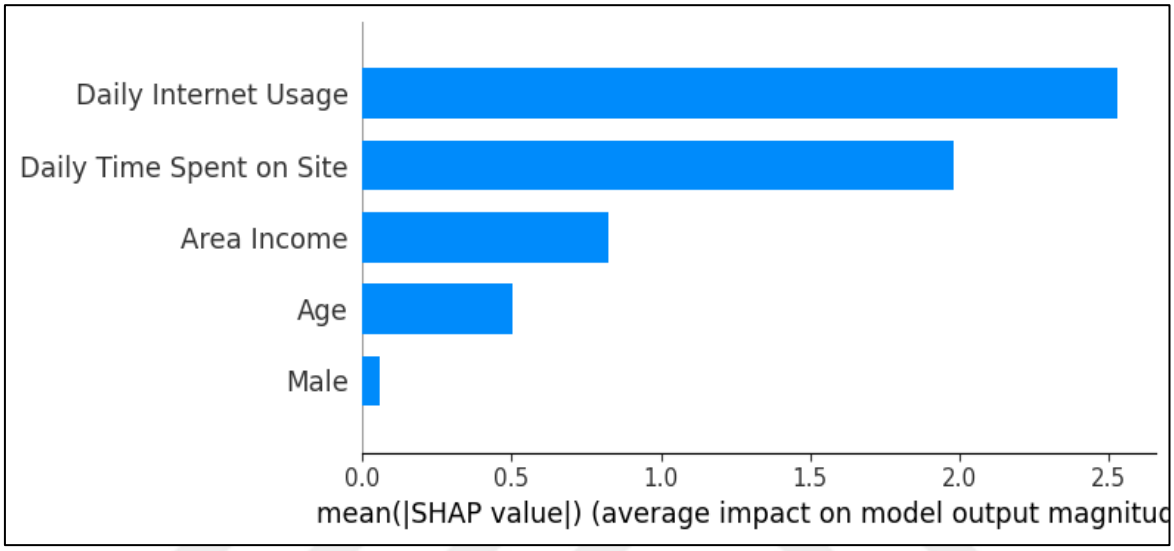

Figure 4.12: Average Impact on Model Output Magnitudes for GB Using SHAP.

4.3.4 Evaluation of the KNN Model

In the evaluation of the KNN model, an ACC of 71% was observed, suggesting a moderate level of effectiveness in classification. To gain further insights into the model's performance, the CM was examined. The matrix revealed that 109 samples were accurately classified as class 0, and 104 were correctly identified as class 1. However, there was a notable number of misclassifications: 37 instances were incorrectly categorized as class 1 when they were actually in class 0, and 50 instances of class 1 were misidentified as class 0. While the model demonstrated a fair level of ACC, these misclassifications indicate areas for potential improvement to enhance its predictive PRE and REC, ensuring a more balanced and reliable classification performance.

Upon examining the CR for the KNN model, a detailed picture of its performance metrics emerges. The model boasts an overall ACC of 83%, indicating a strong degree of reliability in its predictions. When evaluating class-specific metrics, for class 0, the model achieved a PRE of 0.81, signifying that 81% of instances predicted as class 0 were indeed

correct. The REC for class 0 stood at 0.86, meaning that the model correctly identified 86% of all true class 0 instances. The F1-score, which harmoniously balances PRE and REC, was noted at 0.83 for class 0. Similarly, for class 1, the model displayed a commendable PRE of 0.86 and a REC of 0.81, leading to an F1-score of 0.83. This close alignment in F1-S for both classes suggests that the KNN model provides a consistent and balanced performance across the two classes, effectively bridging the gap between PRE and REC.

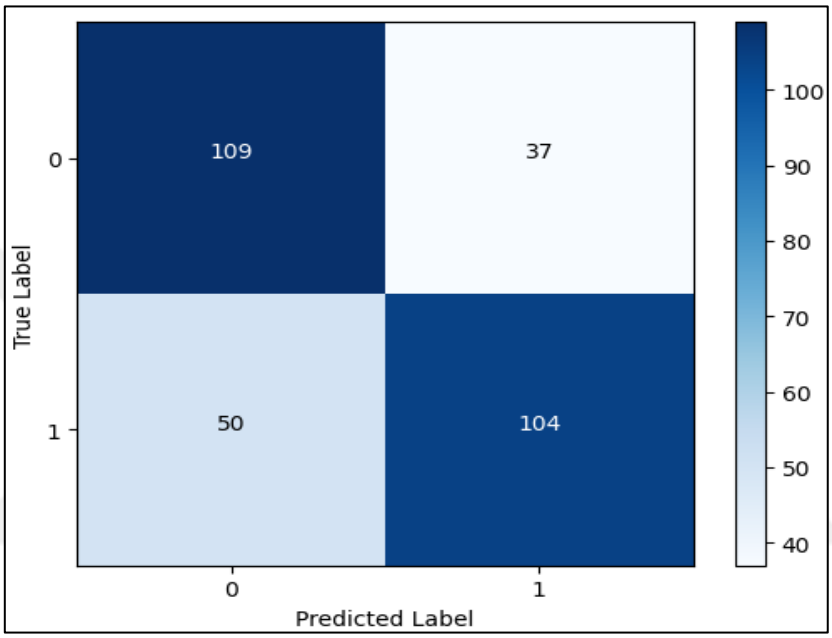

Figure 4.13: KNN Confusion Matrix.

Classification Report:

	precision	recall	f1-score	support
0	0.69	0.75	0.71	146
1	0.74	0.68	0.71	154
accuracy			0.71	300
macro avg	0.71	0.71	0.71	300
weighted avg	0.71	0.71	0.71	300

Figure 4.14: KNN Classification Report.

Figure 4.15 visualizes the feature importance as determined by the LIME methodology when applied to a KNN algorithm. The chart highlights the significance of different attributes in influencing the model's decisions. "Area Income" less than or equal to 47,332.82 dominates with a substantial weight range extending from 0 to 0.49. "Daily

Internet Usage" ranging between 185.45 and 220.06, on the other hand, presents a slight negative impact with weights spanning from -0.1 to 0. "Age" exceeding 41 exhibits a modest influence with weights between 0 and 0.05. "Daily Time Spent on Site" with values between 51.47 and 68.22 has an impact ranging from 0 to 0.03. Lastly, the "Male" feature, when equal to or less than 0, holds the least influence with weights stretching from 0 to 0.01. The figure offers a succinct depiction of each feature's relative importance in the knn model's decision-making process.

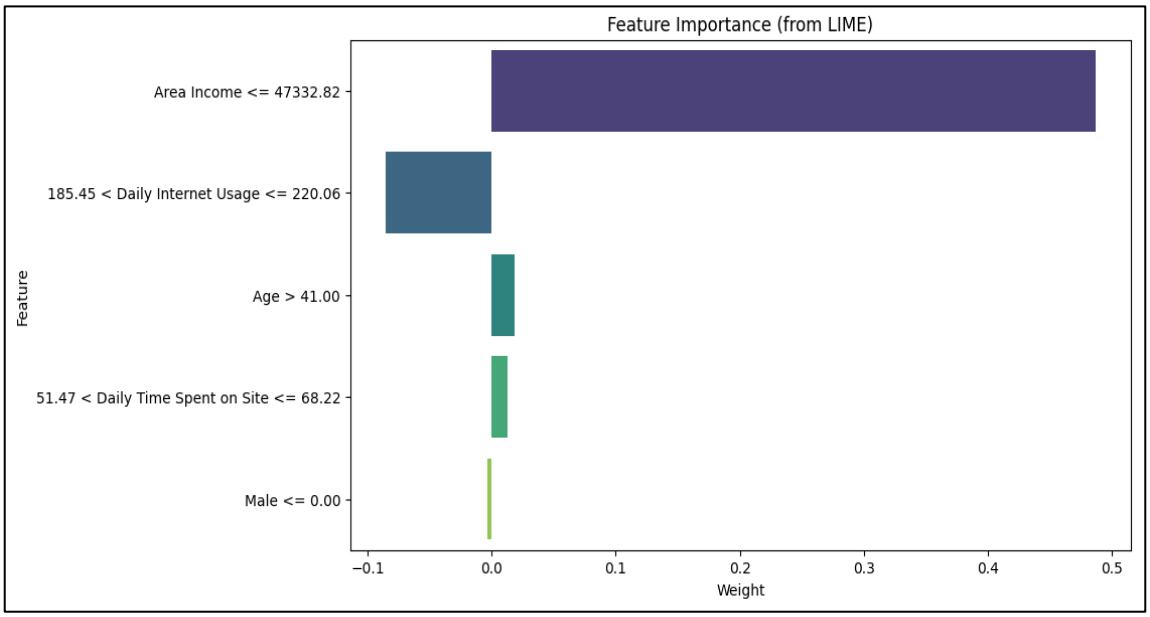

Figure 4.15: Feature Importance Derived from LIME Analysis on a KNN Model.

Figure 4.16 displays a vertical bar chart detailing the average impact on model output magnitudes, as inferred from SHAP values for the KNN method. In this representation, Class 1 is vividly denoted in blue, while Class 0 is depicted in red. The "Area Income" feature for Class 1 holds influence ranging from 0 to 0.25, immediately followed by Class 0's impact extending from 0.25 to 0.5. "Daily Internet Usage" showcases influence for Class 1 between 0 to 0.1 and for Class 0 from 0.1 to 0.2. The "Daily Time Spent on Site" feature has a range of 0 to 0.025 for Class 1, with Class 0 extending its influence from 0.025 to 0.05. For the "Age" attribute, Class 1's impact spans from 0 to 0.005, whereas Class 0's extends from 0.005 to 0.025. Interestingly, the "Male" feature for Class 1 has no discernible impact, being fixed at 0. This bar chart provides a visual differentiation of the features' average impacts on the model's output between the two distinct classes, emphasizing their relative influences.

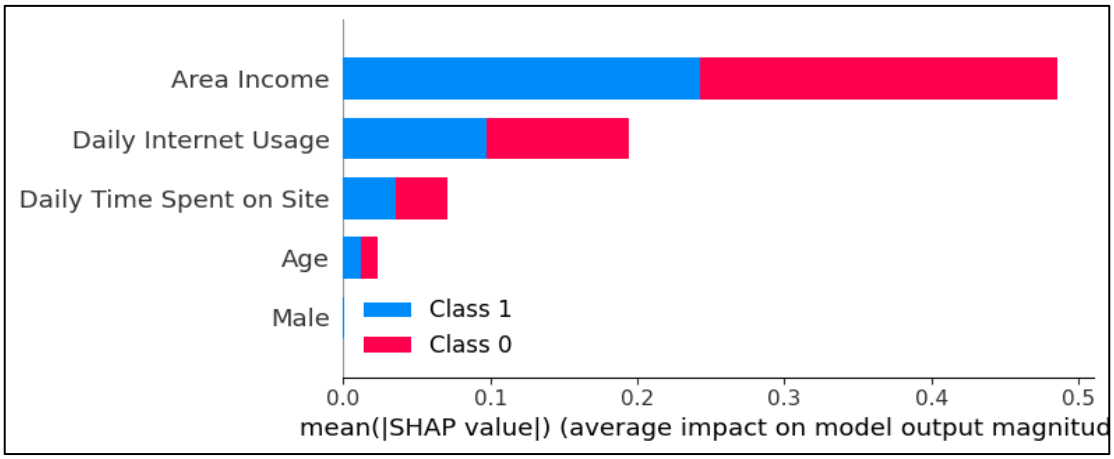

Figure 4.16: Average Impact on Model Output Magnitudes for KNN Using SHAP.

4.3.5 Evaluation of the DT Model

The DT model's evaluation reveals an impressive overall ACC of approximately 94.67%, underscoring its strong capability in classification tasks. Delving deeper into the model's performance metrics through the CM, it becomes evident that the model correctly classified 135 instances as class 0 while accurately determining 149 instances as class 1. Notwithstanding, there were a few misclassifications: 11 instances were erroneously predicted as class 1 when they actually belonged to class 0, and conversely, 5 instances of class 1 were misclassified as class 0. Despite these occasional misclassifications, the high ACC indicates that the DT model exhibits robust predictive power. The model's adeptness in striking a balance between predictive PRE and REC results in consistent and reliable classifications, making it a potent tool for such tasks.

The DT model showcases stellar performance metrics as elucidated by its CR. The model registers an exemplary ACC rate of 95%, signifying its adeptness in making reliable predictions. Delving into the class-specific metrics, class 0 displays a PRE of 0.96, indicating that 96% of the predictions labeled as class 0 were accurate. Its REC stands at 0.92, meaning the model successfully identified 92% of the actual class 0 instances. This leads to a harmonious F1-score of 0.94 for class 0, which perfectly captures the balance between PRE and REC. For class 1, the model boasts a PRE of 0.93 and an impressive REC of 0.97, culminating in an F1-score of 0.95. The parallelism in the F1-S across both classes reaffirms the model's consistent and well-rounded performance. In essence, the DT model not only demonstrates high ACC but also strikes an optimal balance in classifying both classes, cementing its efficacy as a classification tool.

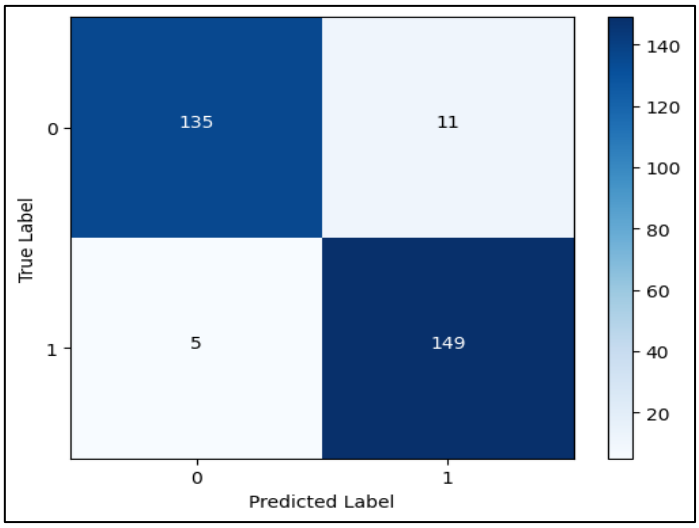

Figure 4.17: DT Confusion Matrix.

Classification Report:	precision	recall	f1-score	support
0	0.96	0.92	0.94	146
1	0.93	0.97	0.95	154
accuracy			0.95	300
macro avg	0.95	0.95	0.95	300
weighted avg	0.95	0.95	0.95	300

Figure 4.18: DT Classification Report

Figure 4.19 offers a concise visualization of feature importance as deduced from the LIME method when applied to a DT model. Within the chart, the significance of various attributes in shaping the model's determinations is evident. "Daily Internet Usage," spanning from 185.45 to 220.06, exhibits a negative influence with weights ranging from -0.3 to 0. "Area Income" capped at 47,332.82 has an impact stretching from 0 to 0.14. The "Age" feature, considering values above 41, carries weights from 0 to 0.09. Notably, "Daily Time Spent on Site" between 51.47 and 68.22 demonstrates a pronounced weight range of 0 to 0.7. Lastly, the "Male" attribute, when equal to or below 0, exerts influence within the 0 to 0.4 range. This figure succinctly encapsulates the relative influence of each feature within the DT model's decision framework.

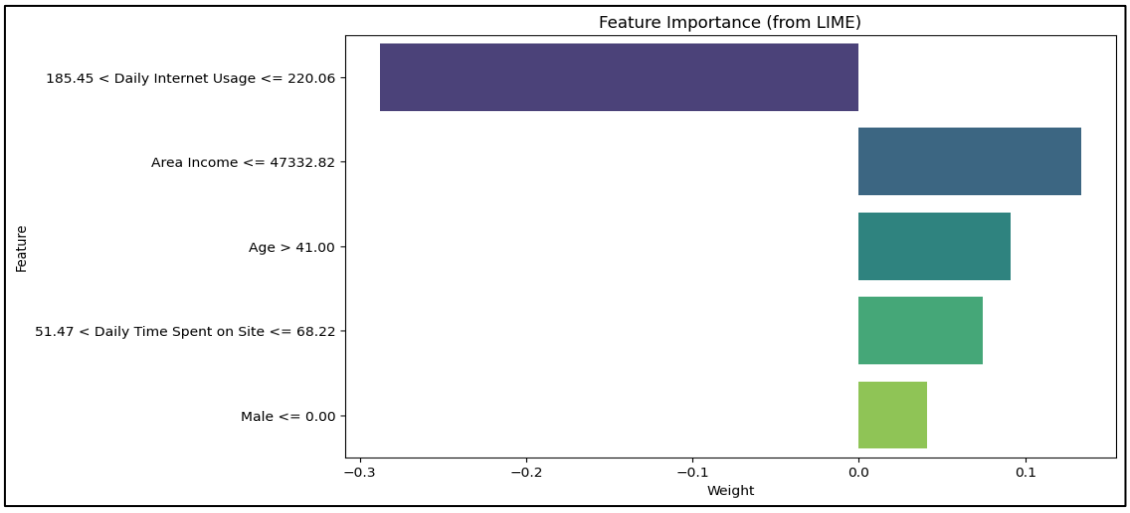

Figure 4.19: Feature Importance Derived from LIME Analysis on a DT Model.

Figure 4.20 portrays a vertical bar chart that elucidates the average impact on model output magnitudes, derived from SHAP values when applied to the DT method. In this illustrative depiction, Class 1 is distinctly color-coded in blue, while Class 0 stands out in red. The feature "Daily Internet Usage" demonstrates an impact ranging from 0 to 0.3 for Class 1, succeeded by Class 0's span stretching from 0.3 to 0.6. For "Daily Time Spent on Site", Class 1's influence is mapped between 0 to 0.19, while Class 0 occupies the space from 0.19 to 0.35. When observing "Area Income", Class 1's impact is confined between 0 to 0.045, immediately followed by Class 0 ranging from 0.045 to 0.09. The "Age" attribute has a Class 1 range from 0 to 0.025, whereas Class 0 extends from 0.025 to 0.05. Notably, the "Male" feature for Class 1 holds a range from 0 to 0.02, while Class 0 slightly reverses, spanning from 0.02 to 0.01. This figure succinctly provides a visual contrast of each feature's average influence on the model's output across the two distinguished classes.

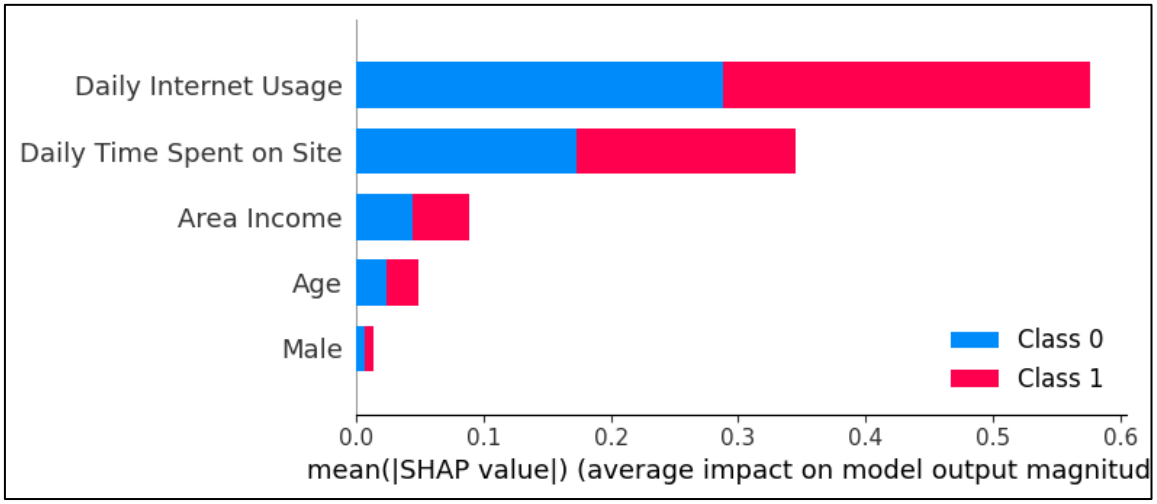

Figure 4.20: Average Impact on Model Output Magnitudes for DT Using SHAP.

4.3.6 Evaluation of the CatBoost Model

The CatBoost model, renowned for its advanced capabilities in handling categorical features, showcases exemplary performance metrics as detailed in the CR. Achieving an overall ACC of 97%, the model stands out in delivering precise and dependable classifications. Breaking down the metrics by class, class 0 has both a PRE and REC of 0.97, resulting in a harmonized F1-score of 0.97. Similarly, for class 1, the model exhibits a PRE, REC, and F1-score, all consistently at 0.97. This parallelism in metrics across both classes emphasizes the model's balanced and accurate performance, ensuring neither class is favored over the other.

The CatBoost model, renowned for its advanced capabilities in handling categorical features, showcases exemplary performance metrics as detailed in the CR. Achieving an overall ACC of 97%, the model stands out in delivering precise and dependable classifications. Breaking down the metrics by class, class 0 has both a PRE and REC of 0.97, resulting in a harmonized F1-score of 0.97. Similarly, for class 1, the model exhibits a PRE, REC, and F1-score, all consistently at 0.97. This parallelism in metrics across both classes emphasizes the model's balanced and accurate performance, ensuring neither class is favored over the other. The CatBoost model, with its coherent PRE, REC, and F1-S, reinforces its prowess as an efficient and reliable ML algorithm for classification tasks.

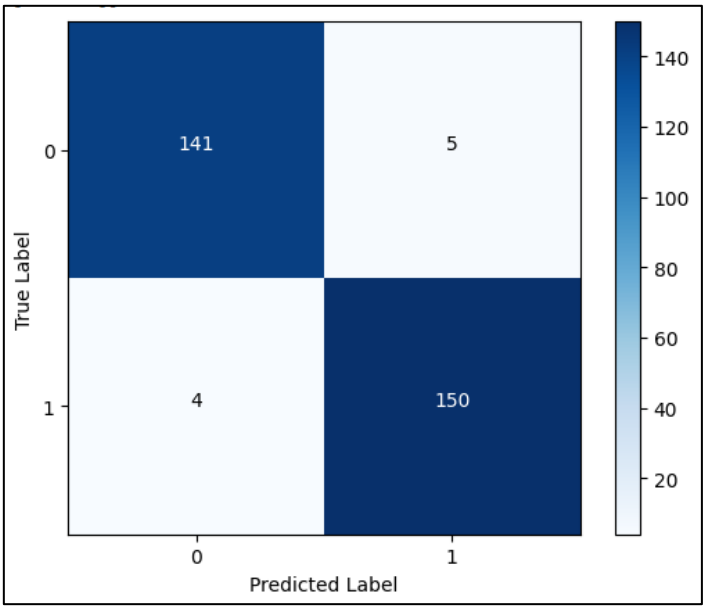

Figure 4.21: Catboost Confusion Matrix.

Classification Report:

	precision	recall	f1-score	support
0	0.97	0.97	0.97	146
1	0.97	0.97	0.97	154
accuracy			0.97	300
macro avg	0.97	0.97	0.97	300
weighted avg	0.97	0.97	0.97	300

Figure 4.22: Catboost Classification Report.

Figure 4.16 provides a clear visualization of feature importance, derived from the LIME technique, when applied to the CatBoost model. The chart highlights the relative significance of distinct attributes in influencing the model's predictions. "Daily Internet Usage" with values between 185.45 and 220.06 holds a negative weight range of -0.3 to 0. "Area Income," not exceeding 47,332.82, exhibits an influence ranging from 0 to 0.16. The "Age" parameter, specifically for values greater than 41, carries an impact ranging between 0 and 0.14. Significantly, "Daily Time Spent on Site" within the bounds of 51.47 and 68.22 manifests a considerable weight stretch from 0 to 0.7. Lastly, the "Male" characteristic, when set to 0 or lower, showcases an impact from 0 to 0.4. This graphical representation succinctly conveys the hierarchy of importance each feature holds within the CatBoost model's decision-making paradigm.

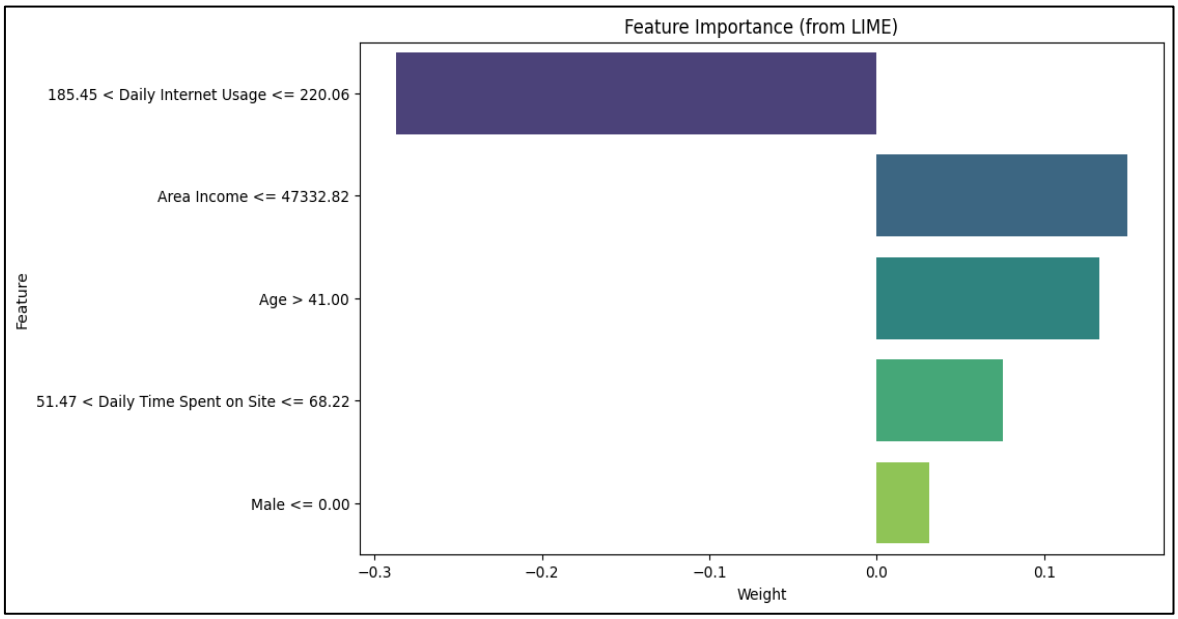

Figure 4.23: Feature Importance Derived from LIME Analysis on a CatBoost Model.

Figure 4.24 visualizes the average impact on model output magnitudes as determined by the SHAP values for the CatBoost method, focusing primarily on Class 1. In the graph, Class 1 is prominently depicted in blue, while Class 0 is shown in red, serving as a contrasting backdrop. "Daily Internet Usage" for Class 1 demonstrates a substantial impact, spanning from 0 to 1.9. Similarly, "Daily Time Spent on Site" exerts influence ranging from 0 to 1.55 for Class 1. The "Area Income" feature for Class 1 showcases an impact extending from 0 to 0.55. "Age" manifests a considerable influence within the range of 0 to 0.49 for Class 1. Lastly, the "Male" attribute presents a more modest range for Class 1, spanning from 0 to 0.1. This figure provides a concise depiction of the relative importance of each feature for Class 1, as assessed by the CatBoost method, within the context of the model's decision-making process.

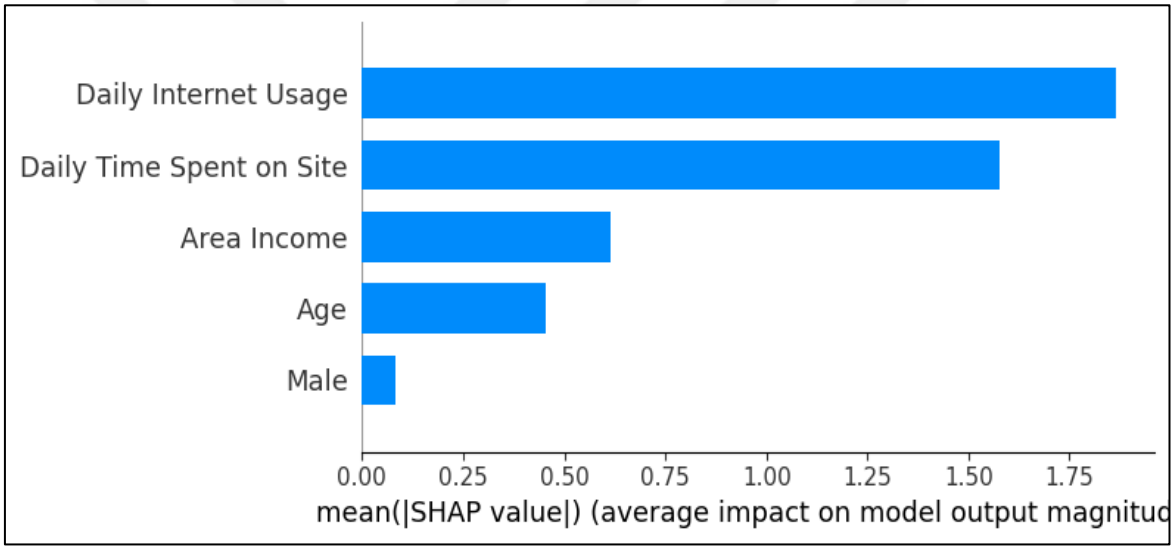

Figure 4.24: Average Impact on Model Output Magnitudes for CatBoost Using SHAP.

4.4 BASE MODELS COMPARISON RESULTS

The Figure 4.25 provides the ACC results for each model before and after parameter tuning.

Before parameter tuning, the RF model achieved an ACC of 97.3%. After parameter tuning, the ACC increased to 97.6%, indicating a slight improvement in performance. For the GB model, the ACC before parameter tuning was 96.6%. After tuning, the ACC remained the same at 96%. The LR model had an ACC of 96.6% before parameter tuning, which increased to 97.3% after tuning. This indicates a notable improvement in ACC for the LR model after the parameter adjustments.

Comparing the ACC results, we observe that both the RF and LR models showed an improvement in ACC after parameter tuning, with the RF model demonstrating a slightly higher ACC than the LR model. On the other hand, the GB model maintained the same ACC level before and after tuning.

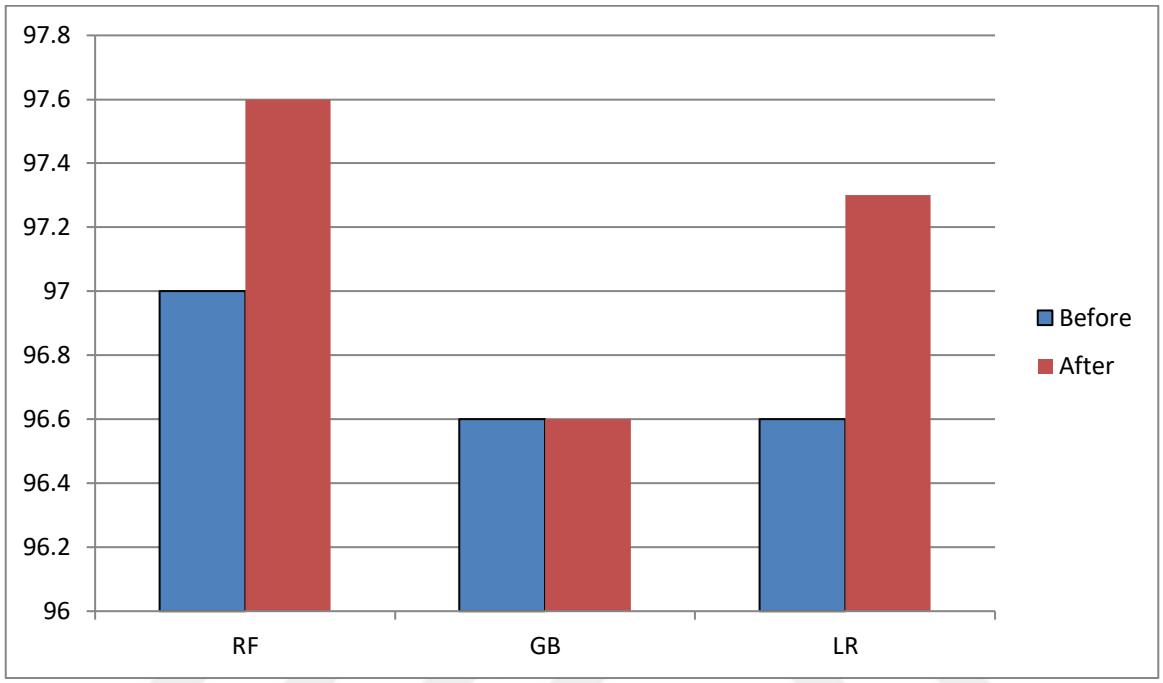

Figure 4.25: Base Models Accuracy Comparison Before and After Parameter Tuning.

4.5 ENSEMBLE MODELS RESULTS

Soft and hard voting are ensemble methods used to combine the predictions of multiple models. Soft voting takes into account the probabilities predicted by each model, while hard voting considers the majority vote of the models. Let's apply soft and hard voting to the three models (RF, GB, and LR) after applying the fine-tuned parameters.

4.5.1 Results using Soft Voting

The ensemble model, constructed through soft voting by combining the predicted probabilities of the fine-tuned RF, GB, and LR models, achieved an ACC of 0.977. The CM reveals that out of the total samples, 142 were correctly classified as belonging to class 0, and 151 were correctly classified as belonging to class 1. There were 4 instances where the ensemble model incorrectly predicted samples as class 1 when they actually belonged to class 0, and 3 instances where samples belonging to class 1 were incorrectly predicted as class 0. The PRE, REC, and F1-score for both classes further validate the performance of

the ensemble model. For class 0, the PRE, REC, and F1-score are 0.98, indicating a high proportion of correctly identified instances and a balanced trade-off between PRE and REC. Similarly, for class 1, the PRE, REC, and F1-score are also 0.98.

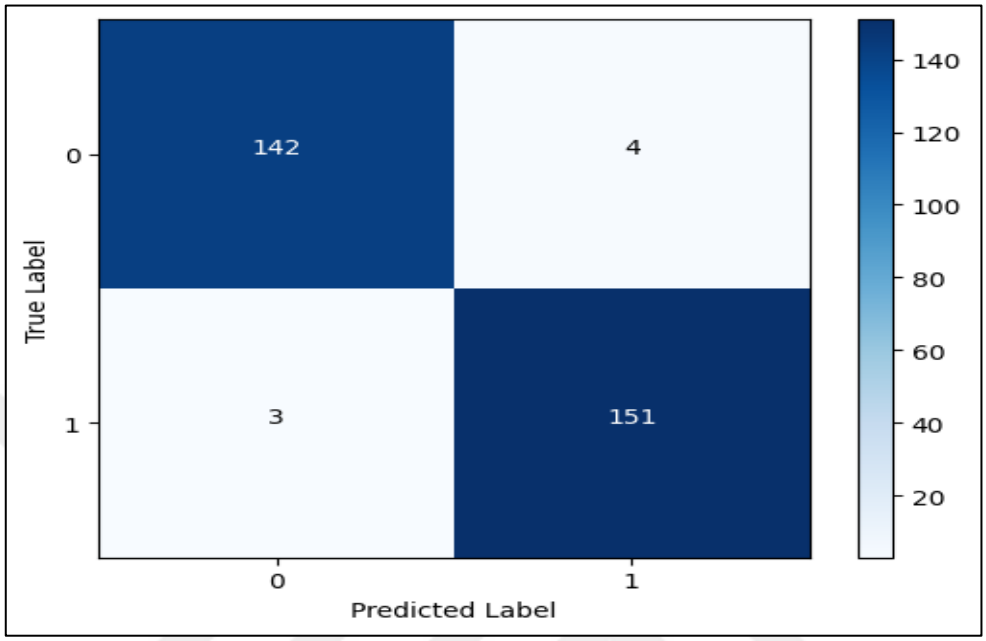

Figure 4.26: Soft Voting Confusion Matrix.

Classification Report:	precision	recall	f1-score	support
0	0.98	0.97	0.98	146
1	0.97	0.98	0.98	154
accuracy			0.98	300
macro avg	0.98	0.98	0.98	300
weighted avg	0.98	0.98	0.98	300

Figure 4.27: Soft Voting Classification Report.

4.5.2 Results using Hard Voting

The ensemble model, created by aggregating the class votes of the fine-tuned RF, GB, and LR models through hard voting, achieved an ACC of 0.973. The CM shows that out of the total samples, 141 were correctly classified as belonging to class 0, while 151 were correctly classified as belonging to class 1. Similarly, there were 5 instances where the ensemble model incorrectly predicted samples as class 1 when they actually belonged to class 0, and 3 instances where samples belonging to class 1 were incorrectly predicted as class 0. The ensemble model's ACC is consistent with the individual models, and the

performance is further validated by the PRE, REC, and F1-score for each class. With a PRE of 0.98 and a REC of 0.97 for class 0, and a PRE of 0.97 and a REC of 0.98 for class 1.

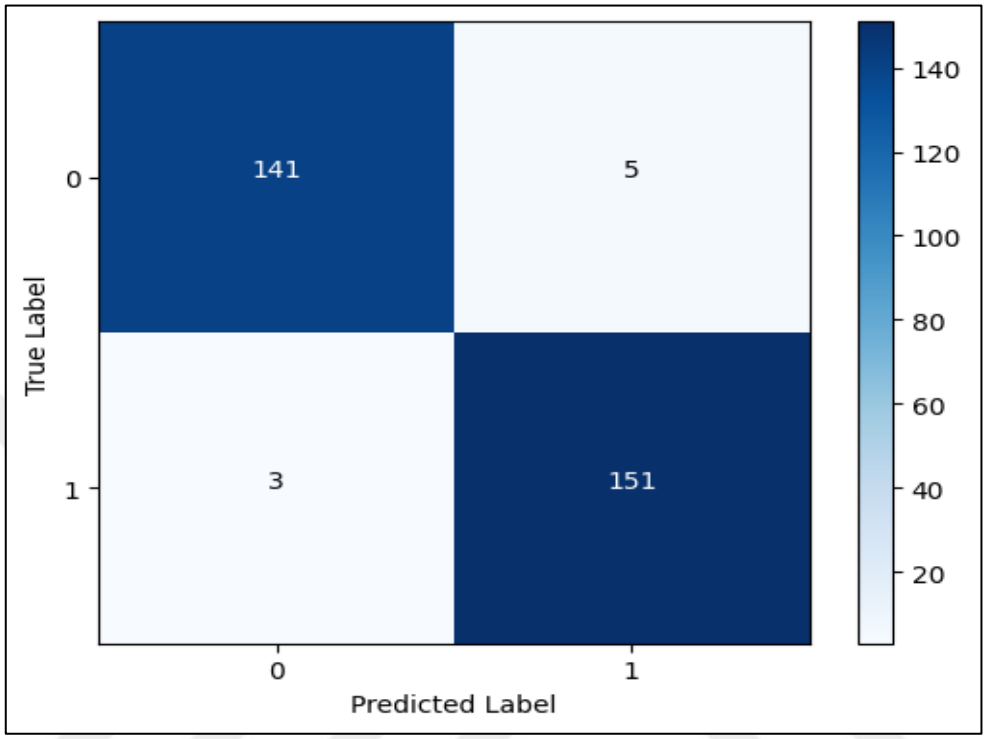

Figure 4.28: Hard Voting Confusion Matrix.

Classification Report:	precision	recall	f1-score	support
0	0.98	0.97	0.97	146
1	0.97	0.98	0.97	154
accuracy			0.97	300
macro avg	0.97	0.97	0.97	300
weighted avg	0.97	0.97	0.97	300

Figure 4.29: Hard Voting Classification Report.

4.5.3 Ensemble Models Comparison

From the above results, it can be observed that both soft voting and hard voting ensembles perform similarly well across all the metrics. They achieve high ACC, PRE, REC, and F1-S for both classes. The soft voting ensemble has a slightly higher ACC of 97.7% compared to the hard voting ensemble's ACC of 97.3%. However, the differences are minimal, suggesting that both ensemble methods effectively combine the predictions of the individual models to produce accurate and reliable results.

4.6 DISCUSSION

The models' performances were evaluated before and after parameter tuning, and ensemble methods were employed to further enhance the predictive capabilities. Initially, the RF achieved an ACC of 97.3%, while the GB and LR models achieved accuracies of 96.6% and 96.6%, respectively. After fine-tuning the parameters, all models demonstrated improved accuracy. The RF and LR models achieved accuracies of 97.7% and 97.3%, respectively, while the GB model maintained its ACC at 96.0%. Subsequently, soft and hard voting ensembles were applied using the fine-tuned models. The soft voting ensemble achieved an ACC of 97.7%, while the hard voting ensemble achieved an ACC of 97.3%. Comparing the ensembles, both exhibited high accuracies and comparable PRE, REC, and F1-S. However, the soft voting ensemble marginally outperformed the hard voting ensemble. Therefore, based on these results, the soft voting ensemble can be considered the best approach, providing the highest ACC and combining the strengths of the individual models effectively.

5. CONCLUSION AND FUTURE WORK

User behavior detection in online advertising is pivotal for crafting targeted and resonant advertisements. Through the nuanced understanding of user behavior, businesses can discern user preferences, interests, and buying tendencies, facilitating the creation of bespoke advertising campaigns. In this research, taking a step forward with the integration of XAI, we crafted a sophisticated methodology for user behavior detection harnessing multiple ML models. We incorporated the RF, GB, and LR , KNN, DT, ChatBoost models, meticulously fine-tuning their configurations to yield optimal outcomes, while also leaning on XAI to shed light on feature importance. To bolster the predictive accuracy, ensemble strategies such as soft voting were also integrated into our approach.

Through our comprehensive analysis, we found that soft voting, which combines the predicted probabilities of the individual models, achieved the best results. The soft voting ensemble exhibited high ACC, PRE, REC, and F1-S, outperforming the other approaches. This highlights the effectiveness of combining the strengths of multiple models to improve user behavior detection in online advertising.

While our research has provided valuable insights into user behavior detection in online advertising, there are several avenues for future exploration. Firstly, expanding the dataset and considering temporal aspects could further refine the understanding of user behavior over time. Furthermore, incorporating additional features, such as demographic information or browsing history, may provide deeper insights into user preferences and improve the PRE of the models. Finally, conducting experiments and validations on larger and diverse datasets from various industries would help to validate and generalize the findings of this research.

REFERENCES

[1] M. B. Albayati and A. M. Altamimi, "An Empirical Study for Detecting Fake Facebook Profiles Using Supervised Mining Techniques," *Informatica*, vol. 43, no. 1, Mar. 2019, doi: 10.31449/inf.v43i1.2319.

[2] C. E. Chibudike, H. Abdu, H. O. Chibudike, O. C. Ngige, O. A. Adeyoju, and N. I. Obi, "Machine Learning - A New Trend in Web User Behavior Analysis," *Int J Comput Appl*, vol. 183, no. 5, pp. 19–25, May 2021, doi: 10.5120/ijca2021921247.

[3] C. E. Chibudike, H. Abdu, H. O. Chibudike, O. C. Ngige, O. A. Adeyoju, and N. I. Obi, "Machine Learning - A New Trend in Web User Behavior Analysis," *Int J Comput Appl*, vol. 183, no. 5, pp. 19–25, May 2021, doi: 10.5120/ijca2021921247.

[4] Statista, "Global advertising spending from 2010 to 2017 (in billion U.S. dollars)." [Online]. Available: https://www.statista.com/statistics/236943/global-advertisingspending/

[5] Statista, "Social media advertising expenditure as share of digital advertising spending worldwide from 2013 to 2017." [Online]. Available: https://www.statista.com/statistics/ 271408/share-of-social-media-in-online-advertising-spending-worldwide/

[6] H. A. Fang *et al.*, "An evaluation of social media utilization by general surgery programs in the COVID-19 era," *The American Journal of Surgery*, vol. 222, no. 5, pp. 937–943, Nov. 2021, doi: 10.1016/j.amjsurg.2021.04.014.

[7] S. Dixon, "Facebook: Global Daily Active Users 2022. Statista. ," www.statista.com/statistics/346167/facebook-global.

[8] S. Dixon, "Facebook: Global Daily Active Users 2022," Statista. [Online]. Available: www.statista.com/statistics/346167/facebook-global

[9] J. Qin, W. Zhang, X. Wu, J. Jin, Y. Fang, and Y. Yu, "User Behavior Retrieval for Click-Through Rate Prediction," in *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, USA: ACM, Jul. 2020, pp. 2347–2356. doi: 10.1145/3397271.3401440.

[10] K. Chaudhary, M. Alam, M. S. Al-Rakhami, and A. Gumaei, “Machine learning-based mathematical modelling for prediction of social media consumer behavior using big data analytics,” *J Big Data*, vol. 8, no. 1, p. 73, Dec. 2021, doi: 10.1186/s40537-021-00466-2.

[11] E. Šoltés, J. Táborecká-petrovičová, and R. Šipoldová, “TARGETING OF ONLINE ADVERTISING,” 2020, doi: 10.15240/tul/001/2020-4-013.

[12] J. J. Almeida, Paulo S and Gondim, “Click fraud detection and prevention system for ad networks,” *Journal of Information Security and Cryptography (Enigma)*, vol. 5, pp. 27--39, 2018.

[13] H. Lipyanina and A. Sachenko, “Decision Tree Based Targeting Model of Customer Interaction with Business Page,” 2020.

[14] E.-A. MINASTIREANU and G. MESNITA, “Light GBM Machine Learning Algorithm to Online Click Fraud Detection,” *Journal of Information Assurance & Cybersecurity*, pp. 1–12, Apr. 2019, doi: 10.5171/2019.263928.

[15] Long Jin, Yang Chen, Tianyi Wang, Pan Hui, and A. V. Vasilakos, “Understanding user behavior in online social networks: a survey,” *IEEE Communications Magazine*, vol. 51, no. 9, pp. 144–150, Sep. 2013, doi: 10.1109/MCOM.2013.6588663.

[16] Mudjahidin, N. L. Sholichah, A. P. Aristio, L. Junaedi, Y. A. Saputra, and S. E. Wiratno, “Purchase intention through search engine marketing: E-marketplace provider in Indonesia,” *Procedia Comput Sci*, vol. 197, pp. 445–452, 2022, doi: 10.1016/j.procs.2021.12.160.

[17] S. Bradshaw, “Disinformation optimised: gaming search engine algorithms to amplify junk news,” *Internet Policy Review*, vol. 8, no. 4, Dec. 2019, doi: 10.14763/2019.4.1442.

[18] A. Kwangsawad, A. Jattamart, and P. Nusawat, “The Performance Evaluation of a Website using Automated Evaluation Tools,” in *2019 4th Technology Innovation Management and Engineering Science International Conference (TIMES-iCON)*, IEEE, Dec. 2019, pp. 1–5. doi: 10.1109/TIMES-iCON47539.2019.9024634.

[19] J. R. Carlson, S. Hanson, J. Pancras, W. T. Ross, and J. Rousseau-Anderson, “Social media advertising: How online motivations and congruency influence perceptions of trust,” *Journal of Consumer Behaviour*, vol. 21, no. 2, pp. 197–213, Mar. 2022, doi: 10.1002/cb.1989.

[20] B. Nyagadza, “Search engine marketing and social media marketing predictive trends,” *Journal of Digital Media & Policy*, vol. 13, no. 3, pp. 407–425, Oct. 2022, doi: 10.1386/jdmp_00036_1.

[21] M. T. P. Adam, J. Krämer, and M. B. Müller, “Auction Fever! How Time Pressure and Social Competition Affect Bidders’ Arousal and Bids in Retail Auctions,” *Journal of Retailing*, vol. 91, no. 3, pp. 468–485, Sep. 2015, doi: 10.1016/j.jretai.2015.01.003.

[22] L. Wei, G. Yang, H. Shoenberger, and F. Shen, “Interacting with Social Media Ads: Effects of Carousel Advertising and Message Type on Health Outcomes,” *Journal of Interactive Advertising*, vol. 21, no. 3, pp. 269–282, Sep. 2021, doi: 10.1080/15252019.2021.1977736.

[23] M. Schreiner, T. Fischer, and R. Riedl, “Impact of content characteristics and emotion on behavioral engagement in social media: literature review and research agenda,” *Electronic Commerce Research*, vol. 21, no. 2, pp. 329–345, Jun. 2021, doi: 10.1007/s10660-019-09353-8.

[24] J. Q. P. Nunes, “How much money should a promotional email marketing campaign really cost?,” Doctoral dissertation, 2014.

[25] L. Kannan and R. Jebakumar, “Public Sender Score System (S3) by ESPs for Email spam mitigation with score management in mobile application.,” 2020.

[26] Y.-C. Hsieh and K.-H. Chen, “How different information types affect viewer’s attention on internet advertising,” *Comput Human Behav*, vol. 27, no. 2, pp. 935–945, Mar. 2011, doi: 10.1016/j.chb.2010.11.019.

[27] D. Lee, K. Hosanagar, and H. S. Nair, “Advertising Content and Consumer Engagement on Social Media: Evidence from Facebook,” *Manage Sci*, vol. 64, no. 11, pp. 5105–5131, Nov. 2018, doi: 10.1287/mnsc.2017.2902.

[28] I. A. Mir and K. Ur REHMAN, “Factors affecting consumer attitudes and intentions toward user-generated product content on YouTube,” *Management & Marketing*, vol. 8, no. 4, 2013.

[29] J. Kang and G. Hustvedt, “Building Trust Between Consumers and Corporations: The Role of Consumer Perceptions of Transparency and Social Responsibility,” *Journal of Business Ethics*, vol. 125, no. 2, pp. 253–265, Dec. 2014, doi: 10.1007/s10551-013-1916-7.

[30] A. Gritckevich, Z. Katona, and M. Sarvary, “Ad Blocking,” *Manage Sci*, vol. 68, no. 6, pp. 4703–4724, Jun. 2022, doi: 10.1287/mnsc.2021.4106.

[31] Y. Chen and Q. Liu, “Signaling Through Advertising When an Ad Can Be Blocked,” *Marketing Science*, vol. 41, no. 1, pp. 166–187, Jan. 2022, doi: 10.1287/mksc.2021.1288.

[32] Y. Chen and Q. Liu, “Signaling Through Advertising When an Ad Can Be Blocked,” *Marketing Science*, vol. 41, no. 1, pp. 166–187, Jan. 2022, doi: 10.1287/mksc.2021.1288.

[33] A. Gritckevich, Z. Katona, and M. Sarvary, “Ad Blocking,” *Manage Sci*, vol. 68, no. 6, pp. 4703–4724, Jun. 2022, doi: 10.1287/mnsc.2021.4106.

[34] B. Nyagadza, “Search engine marketing and social media marketing predictive trends,” *Journal of Digital Media \& Policy*, 2020.

[35] C. Janiesch, P. Zschech, and K. Heinrich, “Machine learning and deep learning,” *Electronic Markets*, vol. 31, no. 3, pp. 685–695, Sep. 2021, doi: 10.1007/s12525-021-00475-2.

[36] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow.* 2022.

[37] L. Breiman, “Bagging predictors,” *Mach Learn*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1007/BF00058655.

[38] D.-S. Cao, Q.-S. Xu, Y.-Z. Liang, L.-X. Zhang, and H.-D. Li, “The boosting: A new idea of building models,” *Chemometrics and Intelligent Laboratory Systems*, vol. 100, no. 1, pp. 1–11, Jan. 2010, doi: 10.1016/j.chemolab.2009.09.002.

[39] K. M. Ting and I. H. Witten, “Stacked Generalization: when does it work?,” 1997.

[40] Y. Guo, X. Wang, P. Xiao, and X. Xu, “An ensemble learning framework for convolutional neural network based on multiple classifiers,” *Soft comput*, vol. 24, no. 5, pp. 3727–3735, Mar. 2020, doi: 10.1007/s00500-019-04141-w.

[41] A. Parmar, R. Katariya, and V. Patel, “A Review on Random Forest: An Ensemble Classifier,” 2019, pp. 758–763. doi: 10.1007/978-3-030-03146-6_86.

[42] H. Fu and K. Qi, “Evaluation Model of Teachers’ Teaching Ability Based on Improved Random Forest with Grey Relation Projection,” *Sci Program*, vol. 2022, pp. 1–12, Feb. 2022, doi: 10.1155/2022/5793459.

[43] A. Natekin and A. Knoll, “Gradient boosting machines, a tutorial,” *Front Neurorobot*, vol. 7, 2013, doi: 10.3389/fnbot.2013.00021.

[44] S. Sperandei, “Understanding logistic regression analysis,” *Biochem Med (Zagreb)*, pp. 12–18, 2014, doi: 10.11613/BM.2014.003.

[45] S. Sun, “Realization Path of College Students’ Network Ideological and Political Teaching System in the New Media Environment,” *Mobile Information Systems*, vol. 2022, pp. 1–9, Sep. 2022, doi: 10.1155/2022/1593350.

[46] L. Peterson, “K-nearest neighbor,” *Scholarpedia*, vol. 4, no. 2, p. 1883, 2009, doi: 10.4249/scholarpedia.1883.

[47] S. B. Kotsiantis, “Decision trees: a recent overview,” *Artif Intell Rev*, vol. 39, no. 4, pp. 261–283, Apr. 2013, doi: 10.1007/s10462-011-9272-4.

[48] C. BAKİR, V. HAKKOYMAZ, B. DİRİ, and M. GÜÇLÜ, “Comparisons on Intrusion Detection and Prevention Systems in Distributed Databases,” *Balkan Journal of Electrical and Computer Engineering*, vol. 7, no. 4, pp. 446–455, Oct. 2019, doi: 10.17694/bajece.605134.

[49] M. R. Zafar and N. Khan, “Deterministic Local Interpretable Model-Agnostic Explanations for Stable Explainability,” *Mach Learn Knowl Extr*, vol. 3, no. 3, pp. 525–541, Jun. 2021, doi: 10.3390/make3030027.

[50] A. Barredo Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Information Fusion*, vol. 58, pp. 82–115, Jun. 2020, doi: 10.1016/j.inffus.2019.12.012.

[51] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable AI: A Review of Machine Learning Interpretability Methods,” *Entropy*, vol. 23, no. 1, p. 18, Dec. 2020, doi: 10.3390/e23010018.