VISVESVARAYA TECHNOLOGICAL UNIVERSITY

‘JNANA SANGAMA’, BELAGAVI-590 018, KARNATAKA

PROJECT REPORT

ON

“PREDICTING AD CLICKS CLASSIFICATION BY USING MACHINE LEARNING”

**SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT
FOR THE AWARD OF THE DEGREE,**

BACHELOR OF ENGINEERING
IN
ARTIFICIAL INTELLIGENCE & DATA SCIENCE

Submitted By

1. **Md Fardeen**
   [1CG21AD022]
2. **Md Tanzil Masud**
   [1CG21AD023]
3. **Mahammad Jaffer Nawaz**
   [1CG22AD403]
4. **Mohammed Shaheed M**
   [1CG22AD404]

Under the guidance of:

Mr.Dharaneshkumar M L, M.Tech.,
Assistant Prof., Dept. of AD,
CIT, Gubbi, Tumakuru.

HOD:

Dr. Gavisiddappa, Ph.D
Prof & Head, Dept. of AD,
CIT, Gubbi, Tumakuru.

Channabasaveshwara Institute of Technology
(NAAC Accredited & ISO 9001:2015 Certified Institution)
NH 206 (B.H. Road), Gubbi, Tumakuru – 572 216. Karnataka.

(Affiliated to Visvesvaraya Technological University, Belagavi & Recognized by AICTE New Delhi)

2024-25

ACKNOWLEDGEMENT

A great deal of time and lot of effort has gone into completing this project report and documenting it. The number of hours spent in getting through various books and other materials related to this topic chosen by us have reaffirmed its power and utility in doing this project.

Several special people have contributed significantly to this effort. First of all, we are grateful to our institution **Channabasaveshwara Institute of Technolog**y, **Gubbi** which provided us an opportunity in fulfilling our most cherished desire of reaching the goal.

We acknowledge and express our sincere thanks to the beloved Director and Principal **Dr. Suresh D S** for his many valuable suggestions and continued encouragement and support in the academic endeavors.

We wish to express our deep sense of gratitude to **Dr. Gavisiddappa**, Professor & Head of the Department of **Artificial Intelligence & Data Science** for all the guidance and who still remains a constant driving force and motivated through innovative ideas with tireless support and advice during the project.

We wish to express our deep sense of gratitude to our guide **Mr.Dharaneshkumar M L** Dept of AD ,CIT, for his meticulous attention to details, which has contributed immeasurably to the quality of the project report

We would express our gratitude towards our parents and friends for their kind cooperation and encouragement which helped us in completion of this project.

Finally, we would like to thank all the teaching and non-teaching staff of Dept of AD, for their cooperation.

Thanking everyone....

Md Fardeen[1CG21AD022]
Md Tanzil Masud [1CG21AD023]
Mohammed Shaheed M [1CG22AD404]
Mahammad Jaffer Nawaz [1CG22AD403]

ABSTRACT

In the rapidly evolving digital marketing landscape, accurately predicting whether a user will click on an advertisement is crucial for maximizing advertising effectiveness and return on investment. This project aims to build a robust machine learning-based classification model to predict ad click behavior using user data such as age, daily time spent on site, daily internet usage, income, and gender. Various algorithms including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Gradient Boosting, and XGBoost were implemented and compared for performance. The model identifies key features influencing ad engagement and provides actionable insights for businesses to optimize their targeting strategies. The results show significant improvements in both click-through rates and advertising profitability, demonstrating the potential of data-driven decision-making in online advertising.

CONTENTS

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

With the rapid growth of digital marketing, online advertisements have become a vital tool for businesses to reach potential customers. However, not all users who see an ad actually click on it. Being able to predict whether a user will click on an advertisement is crucial for optimizing ad campaigns, reducing marketing costs, and increasing the overall effectiveness of advertising strategies.

This task, known as Ad Click-Through Rate (CTR) Prediction, is a binary classification problem where the goal is to predict one of two outcomes: click or no click. Machine Learning (ML) techniques are well-suited for this problem as they can learn complex patterns from historical data and make accurate predictions based on user behavior and contextual features.

In this project, we utilize a range of supervised machine learning algorithms to build and evaluate models for predicting ad clicks. These include:

- Logistic Regression: A statistical model that works well for binary classification problems. It estimates the probability of a user clicking on an ad based on input features.
- Decision Tree: A tree-based model that makes decisions by splitting the data into branches based on feature values. It is easy to interpret and useful for understanding feature importance.
- Random Forest: An ensemble method that builds multiple decision trees and combines their predictions to improve accuracy and reduce overfitting.
- K-Nearest Neighbors (KNN): A simple algorithm that classifies a new instance based on the majority label of its nearest neighbors in the feature space.
- Gradient Boosting: A powerful boosting technique that builds models sequentially, with each new model correcting errors made by the previous ones.
- XGBoost (Extreme Gradient Boosting): An optimized and scalable implementation of Gradient Boosting that is widely used in real-world machine learning competitions for its speed and performance.

The dataset typically includes features such as user demographics (age, gender, location), time of the day, device used, browsing history, and characteristics of the ad itself. By training our models on this data, we aim to accurately predict user interactions with ads.

The performance of each algorithm is evaluated using key metrics such as accuracy, precision, recall, and F1-score. This comparative analysis helps determine the most effective model for real-time ad targeting and personalized marketing.

Ultimately, this project demonstrates how machine learning can be leveraged to make intelligent, data-driven decisions in the advertising industry, leading to better engagement rates, higher conversion, and more efficient use of advertising budgets.

1.1 OBJECTIVES

- Build a machine learning classification model
  To predict whether a user will click on an ad based on user behavior and demographic data.

- Identify key features influencing ad clicks
  To analyze features like Daily Time Spent on Site and Daily Internet Usage that affect ad click predictions.

- Apply and evaluate multiple machine learning algorithms
  To compare models like Logistic Regression, Decision Tree, Random Forest, KNN, Gradient Boosting, and XGBoost for accuracy.

- Perform Exploratory Data Analysis (EDA)
  To understand user behavior and uncover patterns that could improve ad targeting strategies.

- Evaluate model performance
  To measure the accuracy, precision, recall, and F1-score of the models for optimal performance.

- Provide business recommendations for ad targeting
  To suggest personalized content, age-targeted ads, and optimized ad placement strategies based on model insights.

1.2 PROBLEM STATEMENT

In the competitive landscape of digital advertising, businesses face the challenge of effectively targeting users who are most likely to engage with online ads. Traditional methods often result in low click-through rates and suboptimal ad performance. By analyzing user behavior data, such as Daily Time Spent on Site, Daily Internet Usage, Age, and Income level, it becomes evident that certain user segments are more responsive to ads than others. This project aims to address the problem of inefficient ad targeting by developing a machine learning-based classification model that accurately predicts ad click behavior. The goal is to enhance ad personalization, optimize targeting strategies, and ultimately increase both user engagement and advertising profitability.

1.3 SCOPE OF THE PROJECT

The proposed content moderation system is designed to automatically identify user interactions with online advertisements. The scope of the project includes the following:

- Data Collection and Preprocessing : Collect user behavior data, including Daily Time Spent on Site, Daily Internet Usage, Age, and Income. Clean and preprocess the data for modeling.
- Feature Selection and Engineering : Identify and create relevant features that may impact ad click predictions, including demographic and behavioral features.
- Model Development : Implement machine learning algorithms such as Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Gradient Boosting, and XGBoost to predict ad clicks.
- Hyperparameter Tuning : Optimize the performance of each model through hyperparameter tuning to achieve better accuracy and reliability.
- Feature Importance Analysis : Analyze the significance of each feature in predicting ad clicks to provide insights into key user behavior patterns.
- Model Evaluation : Evaluate the models using appropriate metrics (accuracy, precision, recall, F1-score) and select the best-performing model for prediction.

- Business Insights and Recommendations : Provide actionable business recommendations for ad targeting, content personalization, and optimal ad placement based on the model's findings.
- Impact Assessment and Performance Improvement : Measure the improvement in click-through rates and advertising profitability after model implementation, demonstrating the value of machine learning in ad campaigns.

.

CHAPTER 2

LITERATURE SURVEY

- **“Predicting Clicks: Estimating the Click-Through Rate for New Ads” — Agarwal & Chen (2009)**

This paper addresses the challenge of CTR [1] prediction for new ads using a hierarchical model. It improves accuracy by leveraging similarities across ad features and historic data.

- **“Ad Click Prediction: A View from the Trenches” — McMahan et al. (2013)**

Facebook’s deployment of large-scale CTR [2] prediction systems is explained here. The study emphasizes engineering decisions, system latency, and feature selection that impact real-world performance.

- **“Wide & Deep Learning for Recommender Systems” — Cheng et al. (2016)**

The authors propose a hybrid model [3] combining linear (wide) and deep neural networks, allowing the model to memorize frequent patterns while also generalizing to new feature combinations.

- **“Deep Interest Network for Click-Through Rate Prediction” — Zhou et al. (2018)**

DIN [4] learns user interests adaptively from historical behavior, improving CTR prediction by focusing on relevant interaction sequences.

- **“Factorization Machines” — Rendle (2010)**

This paper introduces a machine learning model capable of capturing feature interactions in sparse datasets, which is ideal for CTR [5] prediction tasks.

- **“Field-aware Factorization Machines for CTR Prediction” — Juan et al. (2016)**

FFMs [6] improve upon traditional factorization machines by considering the field (or source) of a feature, leading to improved performance in ad targeting scenarios.

- **“Learning to Rank with Deep Neural Networks for CTR Prediction” — Zhang et al. (2014)**

This study uses DNNs [7] for ranking ads based on CTR probability rather than using standard classification, resulting in better ad placement relevance.

- **“Operation-aware Neural Networks for User Response Prediction” — Yang et al. (2019)**

The ONN [8] model accounts for different user operations (click, view, skip) to enhance user response modeling and CTR prediction, especially in multi-intent environments.

- **“Greedy Function Approximation: A Gradient Boosting Machine” — Friedman (1999)**

This foundational paper introduces [9] gradient boosting, a widely used ensemble method in CTR prediction for its ability to reduce bias and variance.

- **“The Elements of Statistical Learning” — Hastie, Tibshirani, Friedman (2001)**

A comprehensive reference on ML [10] models like logistic regression, decision trees, and boosting, all of which are relevant for building CTR prediction systems.

- **“Bayesian CTR Prediction for Sponsored Search” — Agarwal & Chen (2010)**

This study applies Bayesian hierarchical [11] modeling to handle data sparsity, enabling effective CTR prediction for ads with minimal historical data.

- **“Real-Time Bidding with Multi-Agent Reinforcement Learning” — Zhang et al. (2014)**

Explores bidding optimization in online advertising via [12] reinforcement learning, contributing to more efficient ad placement and improved CTR.

- **“Eye Tracking Study” — Did-it, Enquiro, Eyetools**

Visual behavior [13] data from this study supports that ad position and design significantly affect user engagement and click likelihood.

- **“Logistic Regression and Collaborative Filtering for Sponsored Search” — Bartz et al. (2006)**

Combines logistic regression with collaborative [14] filtering to recommend search terms in sponsored ads, increasing ad visibility and clicks.

- **“Numerical Optimization” — Nocedal & Wright (1999)**

A key resource on optimization algorithms[15] (e.g., BFGS, gradient descent) used to train many CTR models efficiently and effectively.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

3.1.1 Google Ads – Smart Bidding

- Overview: Google Ads' Smart Bidding uses machine learning to optimize bids in real-time for each ad auction.
- How It Works: Models predict the likelihood of a click (CTR) or conversion using contextual signals like device type, location, time of day, browsing behavior, and more.
- Highlight: Its continuous learning mechanism adapts automatically to user behavior and market trends, maximizing return on ad spend (ROAS)

3.1.2 Facebook Ads (Meta Ads) – Click Prediction Engine

- Overview: Meta uses deep neural networks to predict ad clicks across its platforms (Facebook, Instagram, Messenger).
- How It Works: Trains models using user interaction data, demographic attributes, post engagement, and ad features.
- Highlight: Offers one of the most personalized ad experiences with dynamic creative optimization based on predicted engagement.

3.1.3 Amazon Advertising – Sponsored Product Click Predictor

- Overview: Amazon uses ML models to forecast which ads will likely be clicked based on shopping behavior.
- How It Works: Combines purchase history, browsing data, and contextual ad placement signals to predict click probability.
- Highlight: Integrates directly with the buying process, linking ad clicks with actual purchase behavior, enhancing both CTR and ROI.

3.1.4 TikTok Ads – Personalized Ad Delivery

- Overview: TikTok leverages video engagement data to serve personalized ads with high click-through potential.
- How It Works: Uses advanced deep learning (like transformer models) trained on watch history, pause duration, and video interaction.
- Highlight: Delivers short-form video ads that blend natively with user feeds, resulting in high engagement rates.

3.1.5 Criteo Dynamic Retargeting System

- Overview: Criteo uses a machine learning engine to deliver personalized retargeting ads.
- How It Works: Applies large-scale deep learning and field-aware factorization machines to user-level data (like products viewed, time spent).
- Highlight: Excels in e-commerce, with real-time predictions and dynamic ad creatives based on past interactions.

3.2 PROPOSED SYSTEM

3.2.1 Custom Machine Learning Pipeline for Ad Click Prediction

- Overview: A supervised learning-based system designed to classify whether a user will click on an advertisement based on behavioral and demographic data.
- How It Works: The system uses algorithms like Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbors, Gradient Boost, and XGBoost to build a classification model.
- Highlight: Provides flexibility to compare multiple models and choose the best-performing one based on accuracy, precision, recall, and F1-score.

3.2.2 Feature Engineering Based on User Behavior

- Overview: The system focuses on identifying critical features such as Daily Time Spent on Site, Daily Internet Usage, Age, and Income.

- How It Works: Feature importance is extracted using tree-based models and SHAP values to understand which attributes contribute most to ad click behavior.
- Highlight: Helps in interpreting model output and aligning business strategies with real user behavior patterns.

3.2.3 Ad Targeting Personalization Layer

- Overview: Uses classification results to recommend targeted ad segments (e.g., low site time = higher click rate).
- How It Works: The predicted outputs feed into a recommendation engine that groups users based on their likelihood to click, enabling dynamic content delivery.
- Highlight: Increases ad efficiency by delivering the right content to the right audience at the right time.

3.2.4 Model Optimization and Evaluation Dashboard

- Overview: Integrates tools to compare the performance of different models and visualize key metrics like ROC curves and confusion matrices.
- How It Works: Uses cross-validation, grid search, and performance tracking to tune hyperparameters and assess model reliability.
- Highlight: Allows continuous monitoring and updates based on new data, ensuring sustained model performance.

3.2.5 Scalable Framework for Real-World Deployment

- Overview: Designed with modular architecture, making it suitable for integration into real-time ad-serving systems or marketing platforms.
- How It Works: Can be integrated with Flask/Django API, connected to web/app analytics systems for real-time inference.
- Highlight: Provides a scalable solution with potential for deployment in digital marketing, e-commerce, and content platforms.

CHAPTER 4

SYSTEM DESIGN

The design of system deals with how system is developed. It explains the flow of functionalities in brief. The section contains system data flow diagram, flow chart and sequence diagrams described below.

4.1 System Architecture

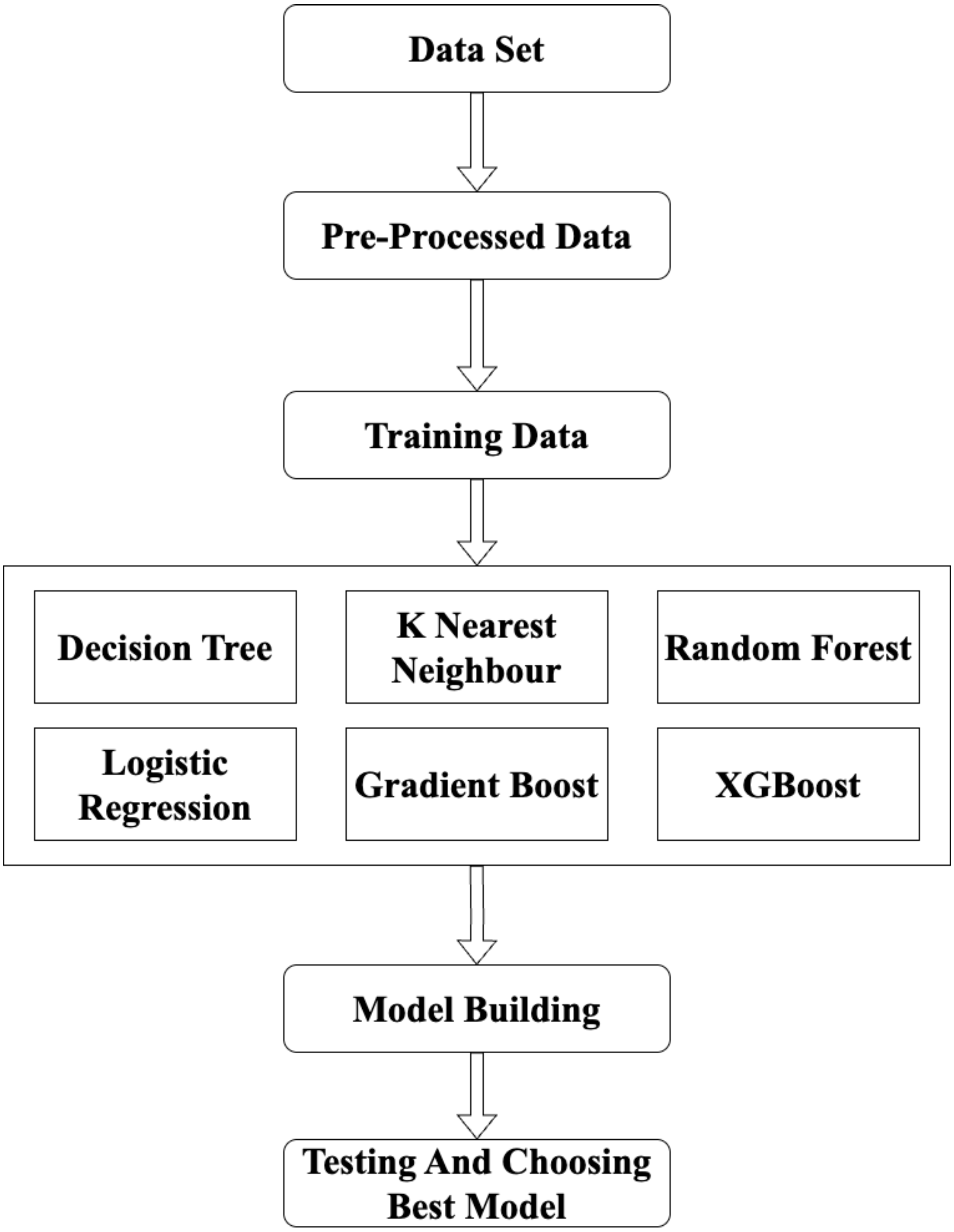


Figure 4.1.1 System Architecture of the ad click prediction system

4.2 Flowchart

Flowcharts are widely used in designing and documenting simple to complex processes across various fields such as computer programming, business operations, and engineering. They help visualize how a system operates or how a task is completed, making it easier to identify redundancies, inefficiencies, or areas for improvement. Common symbols used in flowcharts include ovals for start and end points, rectangles for processes, diamonds for decisions, and arrows to indicate the flow of control. By providing a clear visual structure, flowcharts enhance understanding and communication among team members and stakeholders.

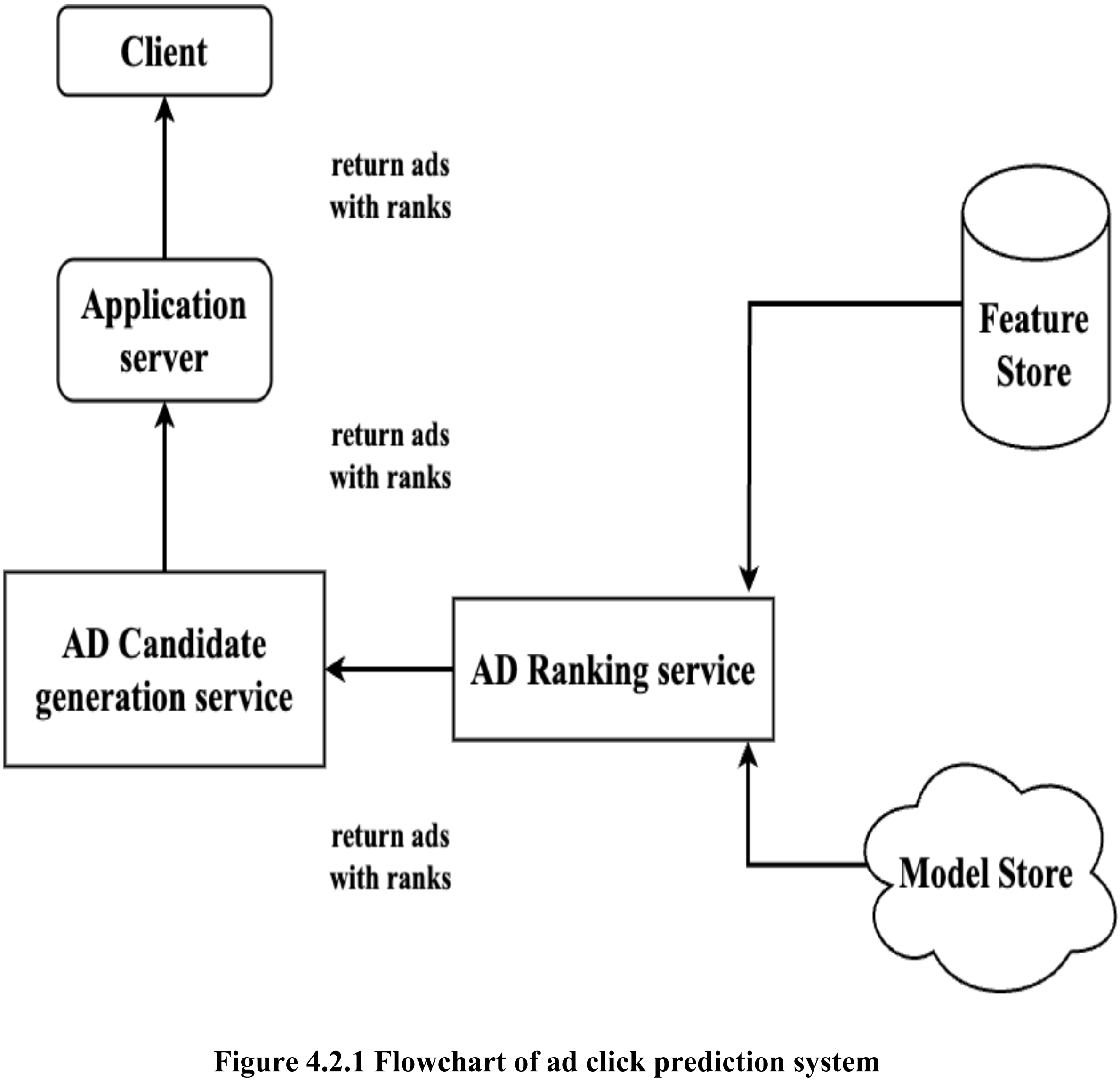


Figure 4.2.1 Flowchart of ad click prediction system

4.3 Sequence Diagram

A sequence diagram is a type of interaction diagram because it describes how and in what order a group of objects works together. It depicts the objects and classesinvolved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the logical view of the system under development. Sequence diagrams are sometimes called event diagrams.

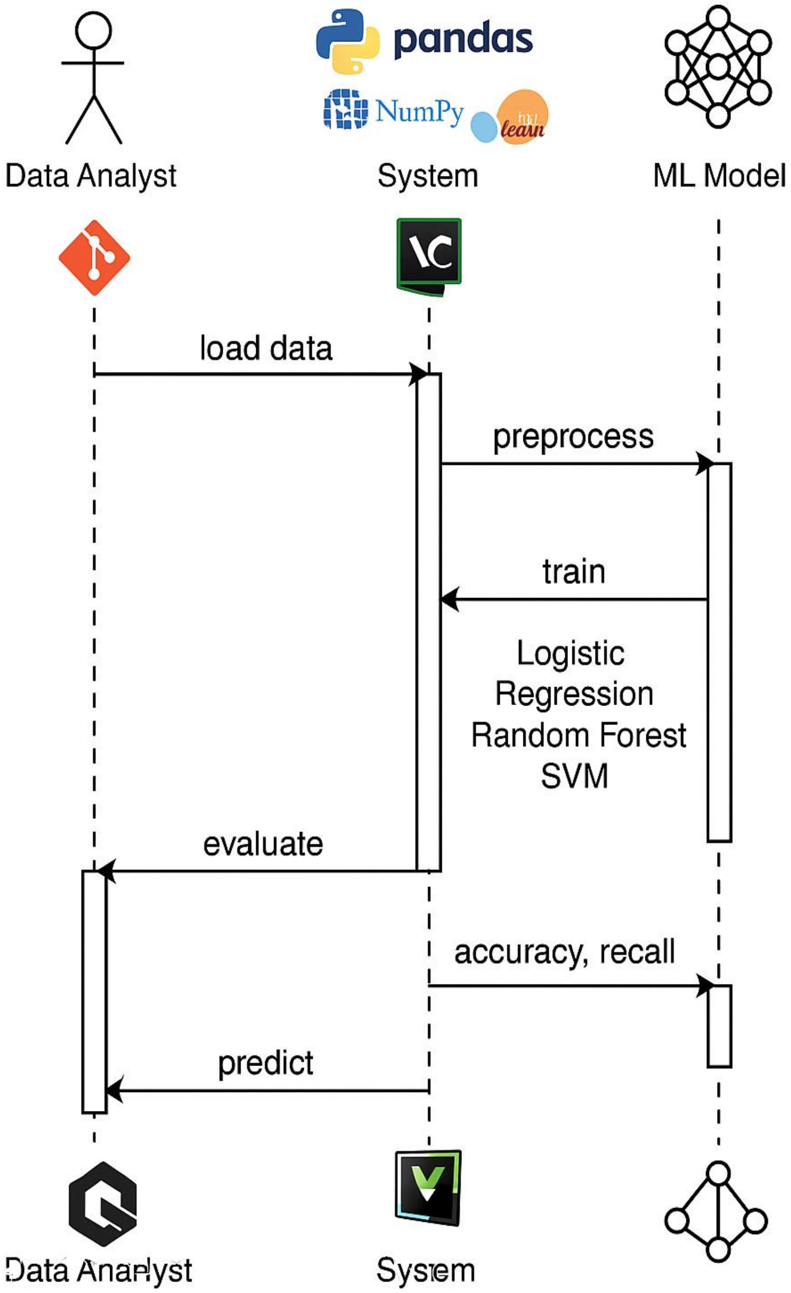


Figure 4.3.1 Sequence diagram

4.4 Use Case Diagram

Use cases are used during the analysis phase of a project to identify system functionality. They separate the system into actors and use cases. Actors represent roles that are played by user of the system. Users may be humans, other computers, or even other software systems.Use case diagrams are used to gather requirements of a system including internal and external influences. These requirements are mostly design requirements. Hence, when a system is analyzed to gather its functionalities, use cases are prepared and actors are identified.

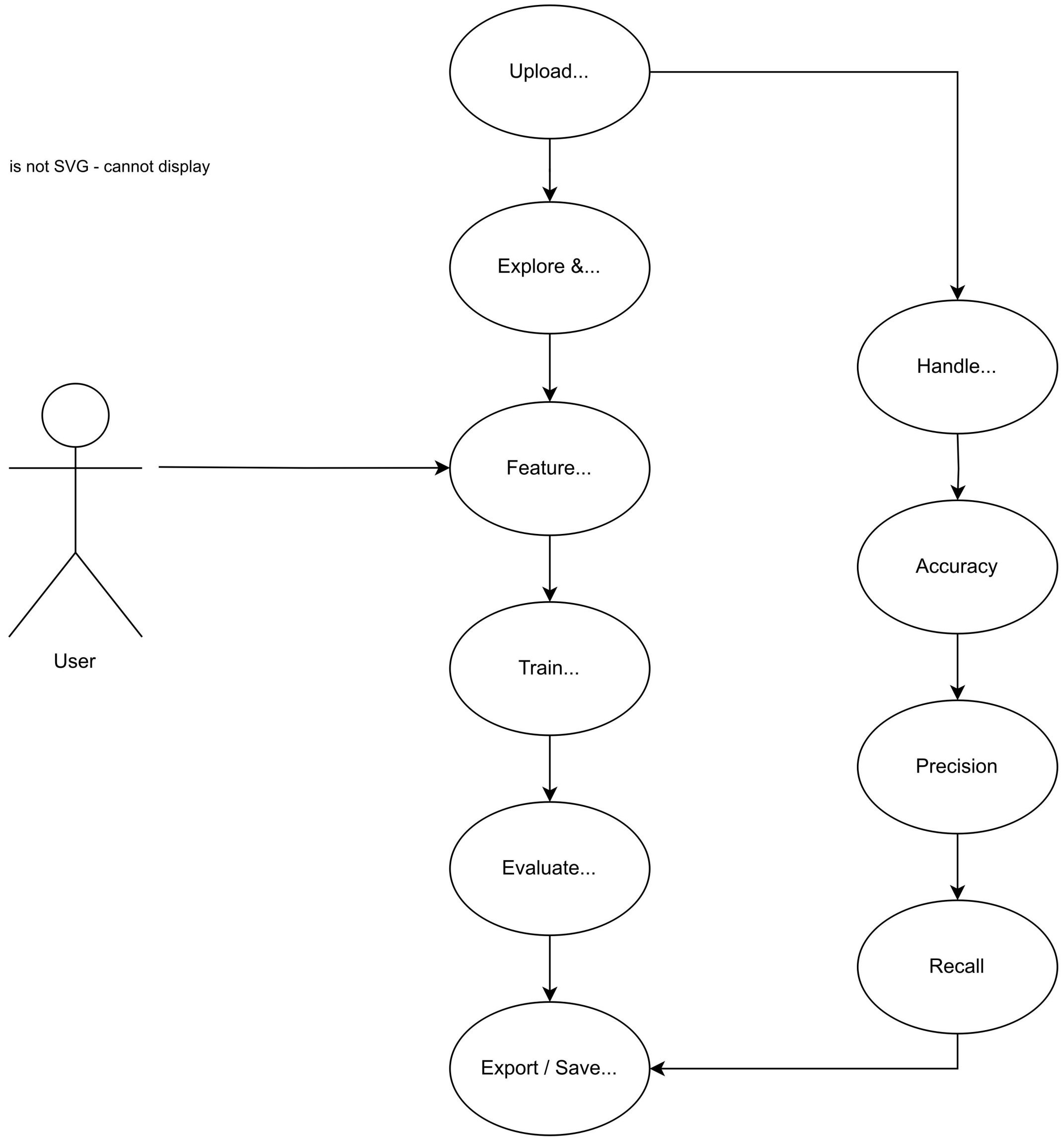


Figure 4.4.1 Use case diagram

CHAPTER 5

IMPLEMENTATION

The implementation of the ad click prediction system involved structured steps, including data preprocessing, model training using multiple machine learning algorithms, and performance evaluation. The goal was to determine the most accurate model for predicting whether a user would click on an advertisement based on demographic and behavioral data.

5.1 Data Collection

The dataset was obtained from Kaggle, containing over 1,000 records with features such as:

- Daily Time Spent on Site
- Age
- Area Income
- Daily Internet Usage
- Gender (Male/Female)
- Clicked on Ad (target variable)

The dataset is multi-label, meaning a single comment can fall under multiple harmful categories. The dataset was obtained from Kaggle and imported into the project using pandas for initial inspection and processing.

This dataset provided a sufficient basis for modeling user interaction behavior and enabled the classification of ad-click events.

5.2 Data Preprocessing

Before training the models, the data underwent several preprocessing steps to ensure its quality and suitability for machine learning:

- **Column Selection**: Non-informative or redundant features such as Ad Topic Line, City, and Timestamp were excluded to reduce noise and simplify the model input.

- **Feature and Target Separation**: The input variables (features) were separated from the target variable, which is the ‘Clicked on Ad’ column.
- **Train-Test Splitting**: The data was split into training and testing sets in an 80:20 ratio. This allowed the models to be trained on one portion of the data and tested on unseen data to evaluate generalization performance.
- **Feature Scaling**: As the dataset contained features with varying scales (e.g., income and age), standardization was performed. This ensured that the features were on a similar scale, improving the performance of algorithms that are sensitive to feature magnitude.

5.3 Feature Extraction

Feature extraction involved selecting key attributes from the dataset that influence ad-click behavior, including Daily Time Spent on Site, Age, Area Income, Daily Internet Usage, and Gender. The categorical feature (Gender) was converted to numerical format, and all numerical features were scaled to ensure consistency. The target variable ‘Clicked on Ad’ was binary, indicating whether a user clicked the ad. These processed features were then used to train the machine learning models effectively.

5.4 Model Training

The processed dataset was split into training and testing sets to evaluate model performance. Multiple machine learning algorithms were trained using the training data, including Logistic Regression, Decision Tree, Random Forest, K-Nearest Neighbour, Gradient Boost, and XGBoost. Each model was trained to learn patterns between user features and the target variable (Clicked on Ad). The goal was to identify which model provided the most accurate and reliable predictions on unseen data.

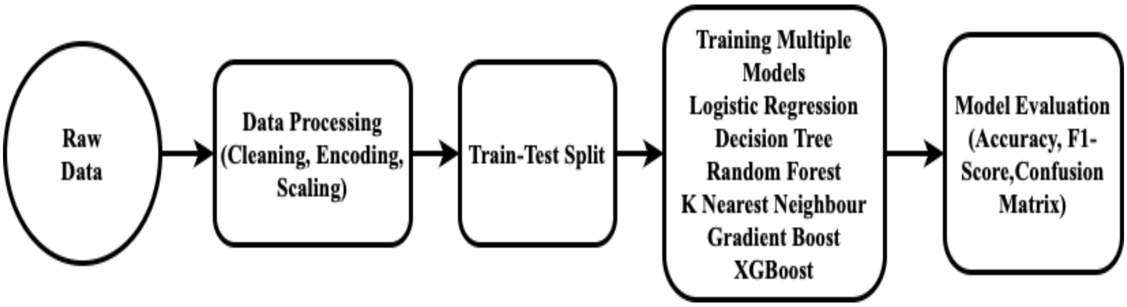


Figure 5.4.1 Model Architecture

Step-by-step explanation of the model architecture:

1. Raw Dataset Collection

The process begins with collecting the dataset, which contains various user-related features such as age, daily internet usage, income, and whether the user clicked on an ad. This raw data forms the foundation of the machine learning pipeline.

2. Data Preprocessing

Before training, the raw data must be cleaned and prepared. This involves:

- Removing or handling any missing values.
- Encoding categorical variables (like gender) into numerical format.
- Scaling features so that algorithms sensitive to numerical ranges (e.g., KNN, Logistic Regression) perform better.
- Ensuring the data is in a format suitable for training.

3. Train-Test Split

The dataset is split into two parts:

- Training set (typically 80%) to train the machine learning models.
- Test set (typically 20%) to evaluate how well the trained models perform on unseen data.
- This step ensures that the model is not just memorizing the data but is capable of generalizing to new inputs.

4. Model Selection and Training

Multiple machine learning models are selected and trained on the training data. In this project, the following models are used:

- **Logistic Regression** – for its simplicity and interpretability.
- **Decision Tree** – for modeling non-linear decision boundaries.
- **Random Forest** – for handling overfitting and improving accuracy using an ensemble of trees.

- **K-Nearest Neighbour** – for instance-based learning based on similarity.
- **Gradient Boost** – for high-performance predictions using boosting.
- **XGBoost** – a powerful and optimized boosting algorithm used for structured data.

Each model learns from the training data to understand the relationship between features (like age or internet usage) and the output (whether the ad was clicked).

5. Model Evaluation

After training, the models are tested using the test data. Performance is measured using metrics such as:

- **Accuracy** – how often the model predicts correctly.
- **F1-Score** – balance between precision and recall.
- **Confusion Matrix** – detailed breakdown of correct and incorrect predictions.

These metrics help compare different models and identify the best-performing one for deployment.

5.5 Evaluation

To assess the performance of each classification model, several evaluation metrics were used:

- Accuracy: The proportion of total correct predictions out of all predictions made.
- Precision: The proportion of correctly predicted positive observations out of all predicted positives.
- Recall: The proportion of correctly predicted positives out of all actual positives.
- F1-Score: The harmonic mean of precision and recall, providing a balance between the two.
- Confusion Matrix: A matrix that breaks down the number of true positives, true negatives, false positives, and false negatives, giving a complete picture of prediction results.

5.6 System Integration and Deployment

Although the primary focus of the project was on model training and evaluation, it was important to consider deployment for real-world usability. The trained model could be integrated into a web-based or mobile system where user data is collected through forms or logs and fed into the model for real-time ad click predictions.

- Deployment could be done using platforms such as:
- Flask/Django (Python-based APIs)
- Streamlit (for quick deployment and UI)
- Docker (for containerizing the application)
- Cloud platforms (AWS, GCP, or Heroku) for hosting and scalability

This ensures the model is accessible to marketing teams or advertising platforms for decision-making.

5.7 Machine Learning Algorithms

This section outlines each algorithm used in this project and explains their working principles and characteristics:

5.7.1 Logistic regression: Logistic Regression is a statistical model used for binary classification problems. It predicts the probability of the target variable belonging to a certain class using a sigmoid function. It assumes a linear relationship between the input variables and the log-odds of the target variable. Logistic Regression is simple, fast, and interpretable, making it ideal as a baseline model for classification tasks like predicting ad clicks.

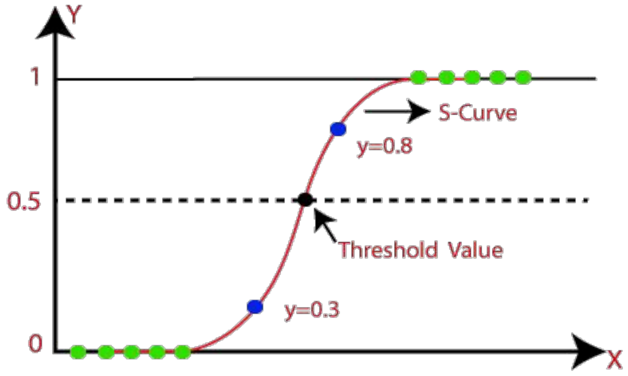


Fig. 5.7.1.1 Logistic Regression

5.7.2 Random Forest: Random Forest is an ensemble learning method that constructs multiple decision trees and combines their results to improve accuracy and prevent overfitting. Each tree is trained on a different subset of the data, and the final prediction is made by averaging the outputs (in regression) or taking a majority vote (in classification). It is robust, handles missing data well, and performs feature importance analysis.

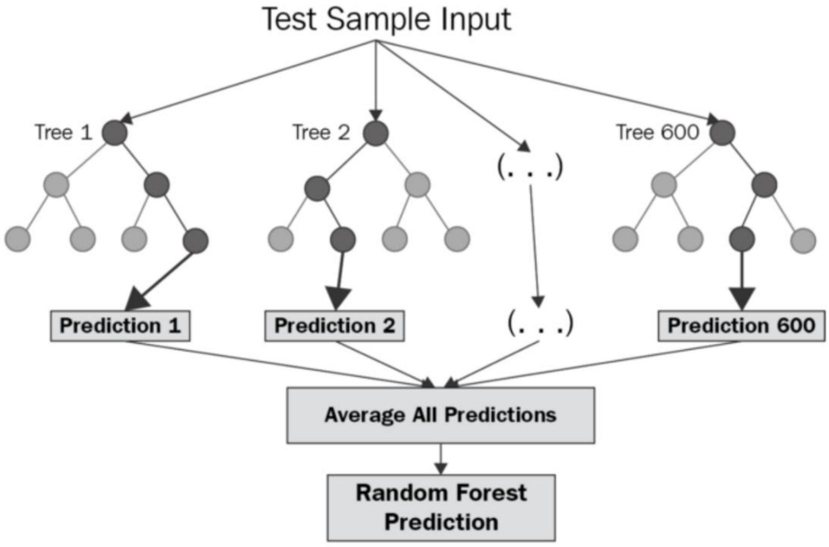


Fig. 5.7.2.1 Random Forest

5.7.3 Decision Tree: A Decision Tree is a non-linear model that splits the data into branches based on feature thresholds. Each node in the tree represents a decision rule, and each leaf represents an outcome. It works well with both numerical and categorical data and is easy to interpret. However, decision trees can overfit the training data if not properly pruned or constrained.

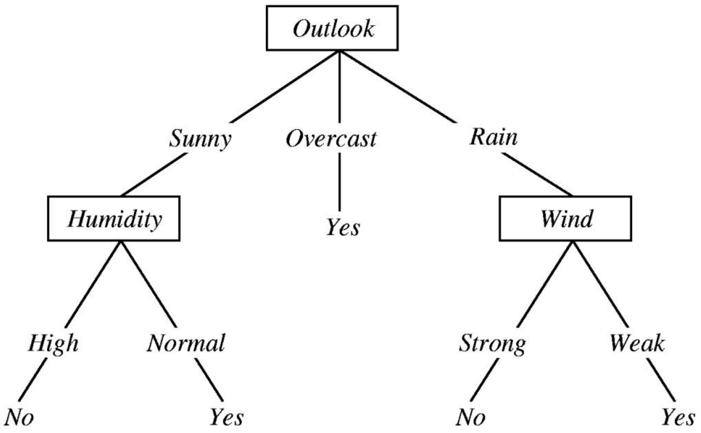


Fig. 5.7.3.1 Decision Tree

5.7.4 K Nearest Neighbour (KNN): KNN is a lazy learning algorithm that stores all training data and makes predictions based on the k closest data points in the feature space. The label is determined by the majority vote among the neighbors. It is simple and effective for small datasets but can become computationally expensive and less accurate on larger or high-dimensional data.

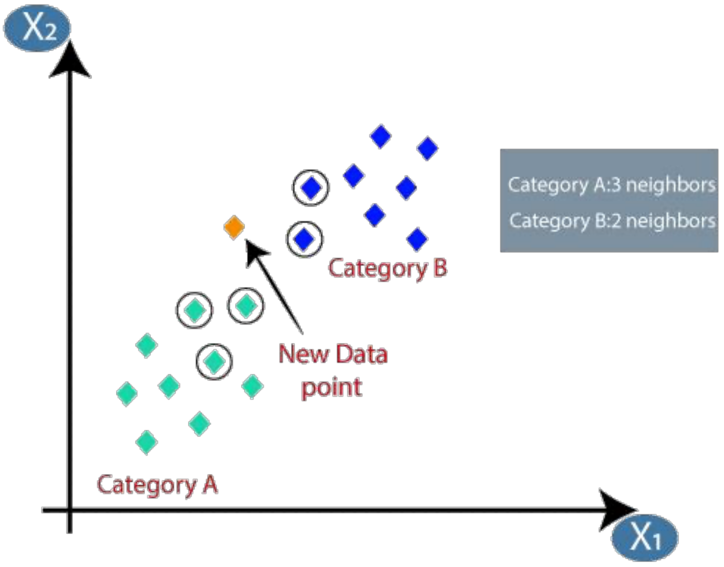


Fig. 5.7.4.1 K Nearest Neighbour (KNN)

5.7.5 Gradient Boost: Gradient Boosting is a sequential ensemble technique where new models are built to correct the errors made by previous ones. It optimizes a loss function using gradient descent. Each tree added to the ensemble focuses on the residual errors of the previous trees. Gradient Boosting generally achieves high performance, especially on structured data, but can be sensitive to noisy data and overfitting without proper tuning.

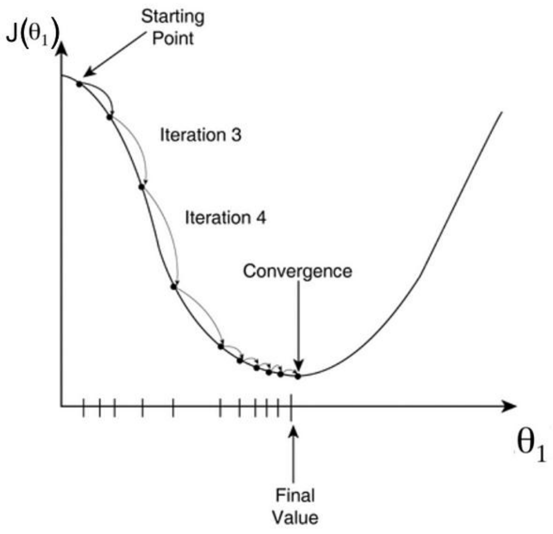


Fig. 5.7.5.1 Gradient Boost

5.7.6 XGBoost: XGBoost (Extreme Gradient Boosting) is an optimized implementation of gradient boosting that provides high performance and speed. It includes regularization techniques to avoid overfitting and uses efficient system optimization like parallel processing. It is often the go-to algorithm for winning machine learning competitions due to its balance of speed, accuracy, and flexibility. In this project, XGBoost showed strong predictive ability and was one of the best-performing models.

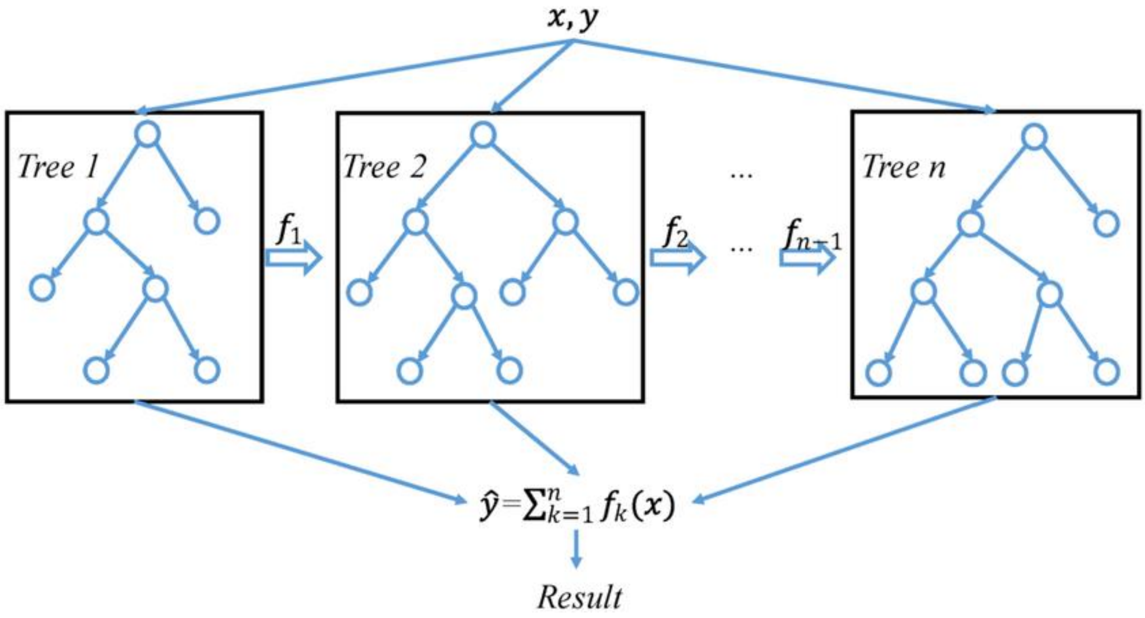
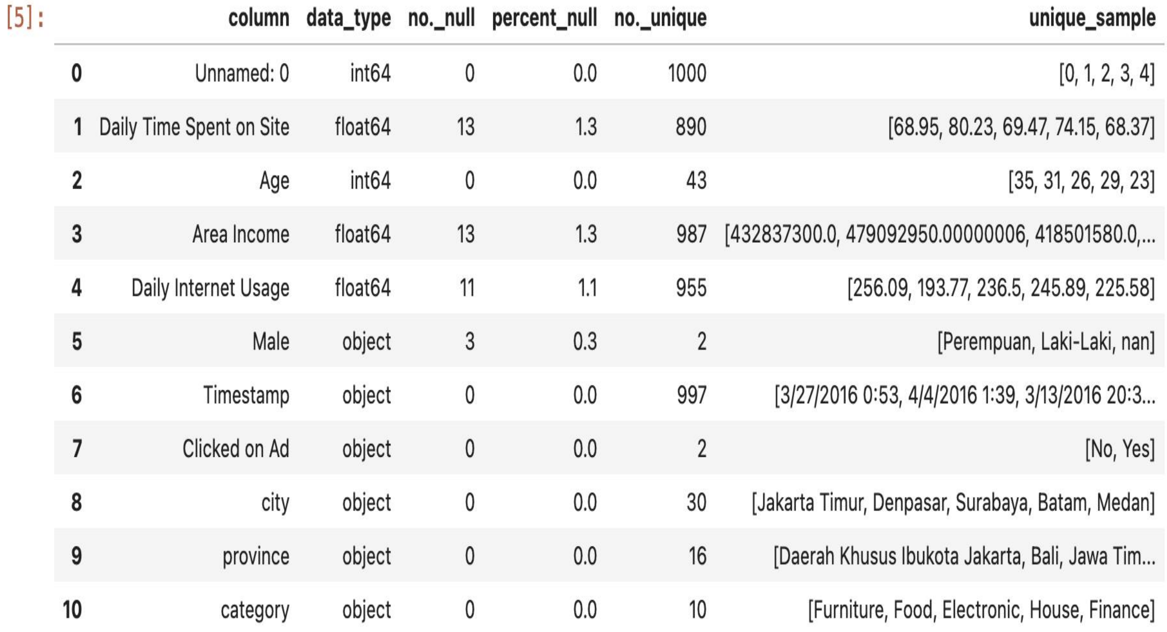


Fig. 5.7.6.1: XGBoost

CHAPTER 6

RESULT



[5]:

	column	data_type	no._null	percent_null	no._unique	unique_sample
0	Unnamed: 0	int64	0	0.0	1000	[0, 1, 2, 3, 4]
1	Daily Time Spent on Site	float64	13	1.3	890	[68.95, 80.23, 69.47, 74.15, 68.37]
2	Age	int64	0	0.0	43	[35, 31, 26, 29, 23]
3	Area Income	float64	13	1.3	987	[432837300.0, 479092950.00000006, 418501580.0,...
4	Daily Internet Usage	float64	11	1.1	955	[256.09, 193.77, 236.5, 245.89, 225.58]
5	Male	object	3	0.3	2	[Perempuan, Laki-Laki, nan]
6	Timestamp	object	0	0.0	997	[3/27/2016 0:53, 4/4/2016 1:39, 3/13/2016 20:3...
7	Clicked on Ad	object	0	0.0	2	[No, Yes]
8	city	object	0	0.0	30	[Jakarta Timur, Denpasar, Surabaya, Batam, Medan]
9	province	object	0	0.0	16	[Daerah Khusus Ibukota Jakarta, Bali, Jawa Tim...
10	category	object	0	0.0	10	[Furniture, Food, Electronic, House, Finance]

Fig 6.1 Descriptive Statistics

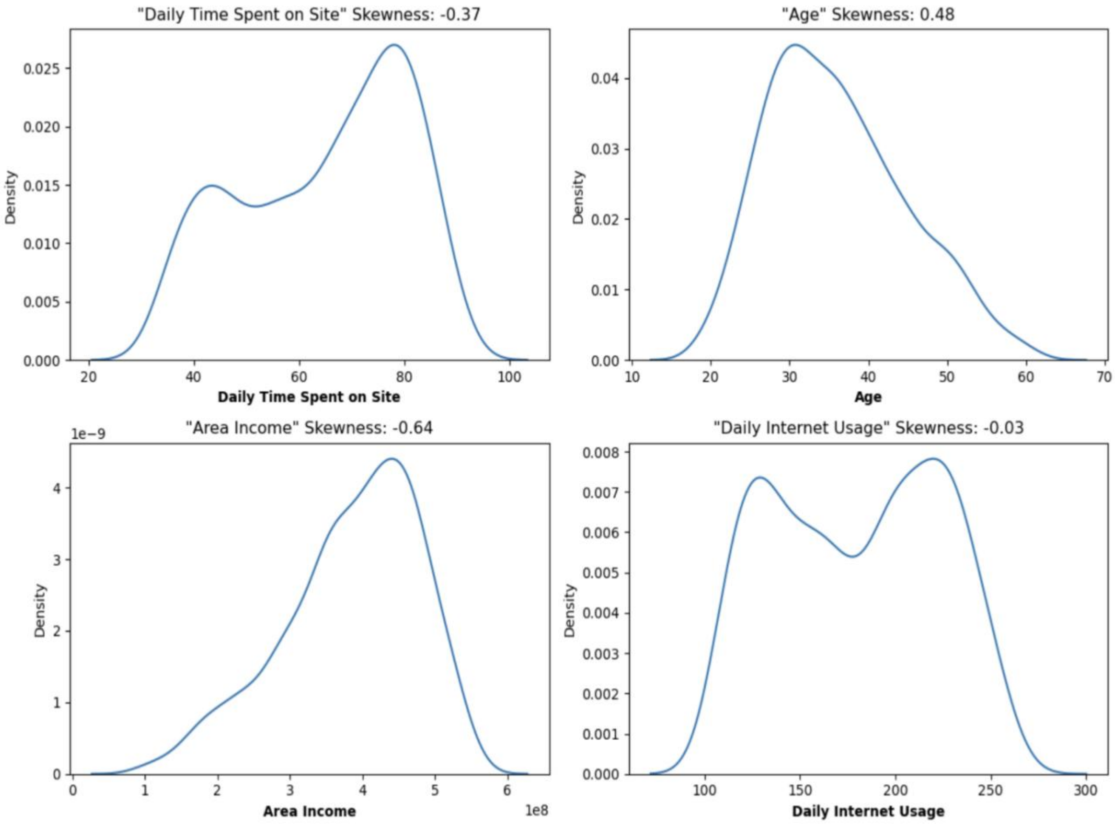
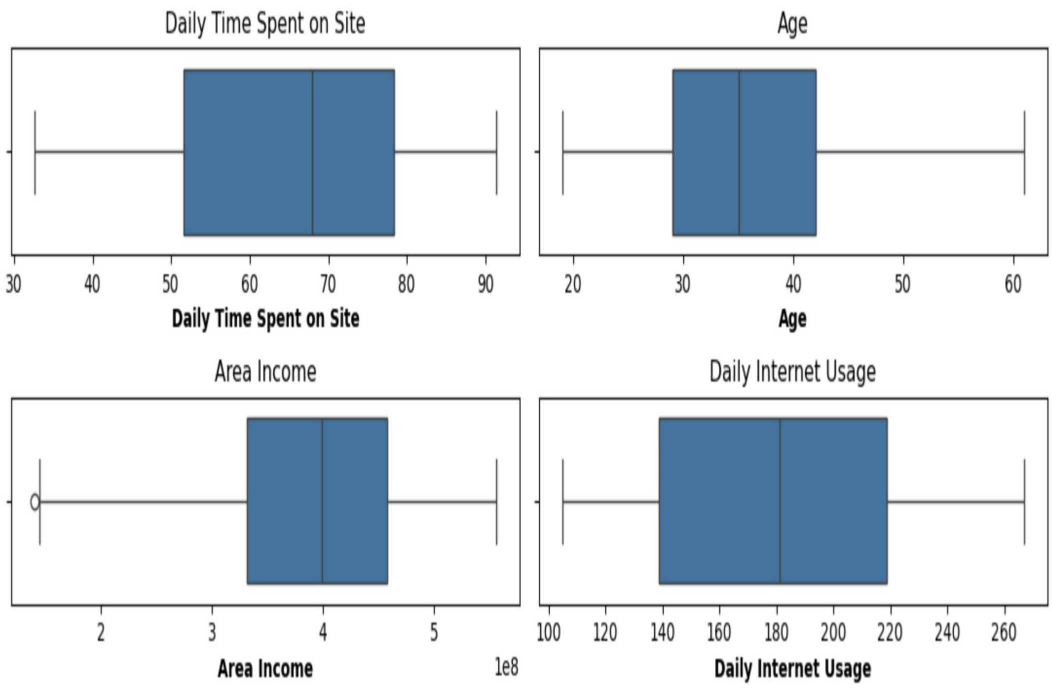


Fig 6.2 Kernal Density Estimate of Numerical Features



As can be seen above, `Area Income` has no glaring outliers remaining.

Fig 6.3 Boxplot Numerical Features

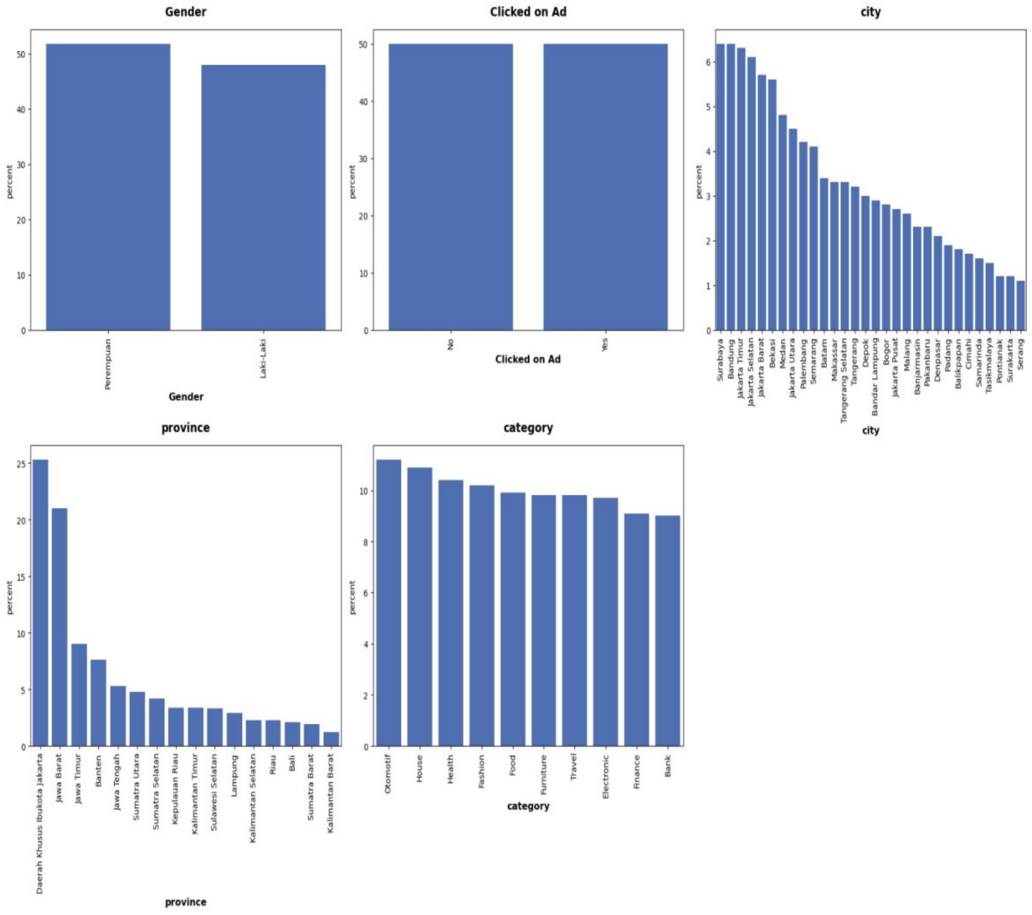


Fig 6.4 Count plot Numerical Features

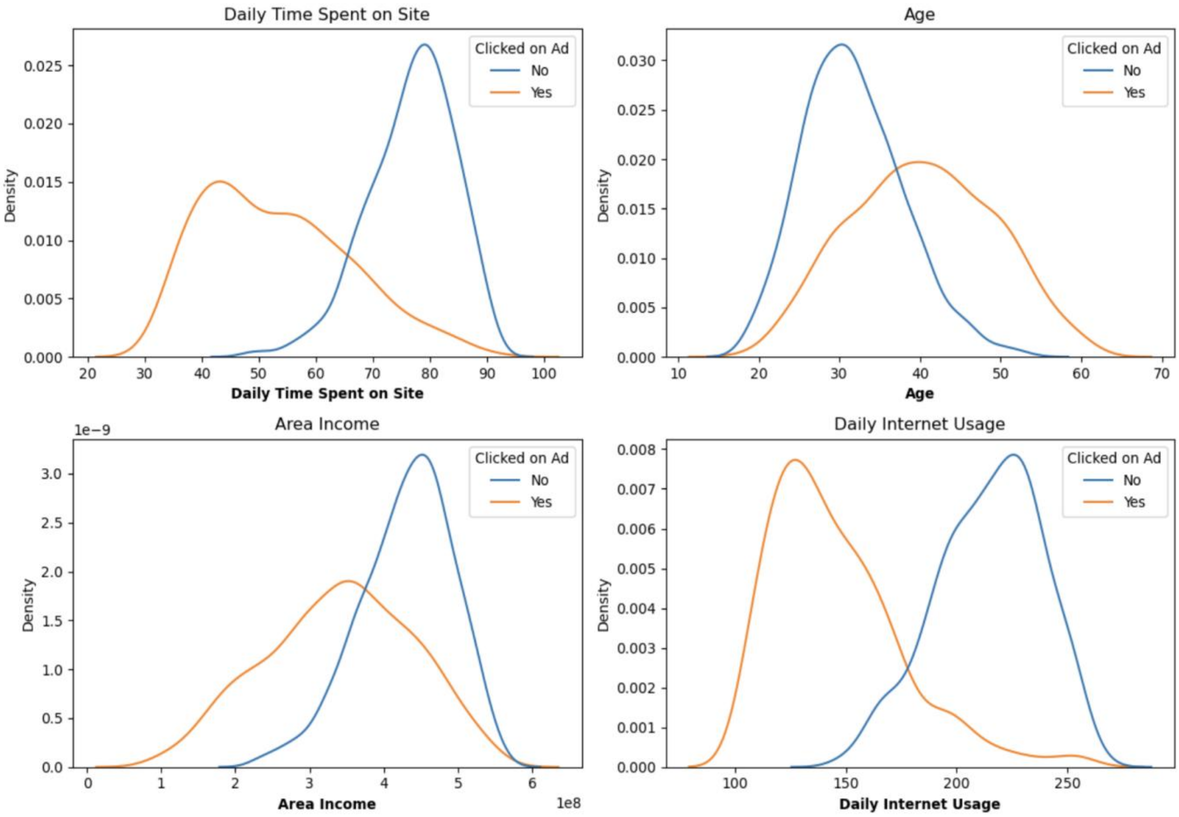


Fig 6.5 Kernal Density Estimate Plot of Numerical Features Between Users That Clicked and Didn't Click On Ad

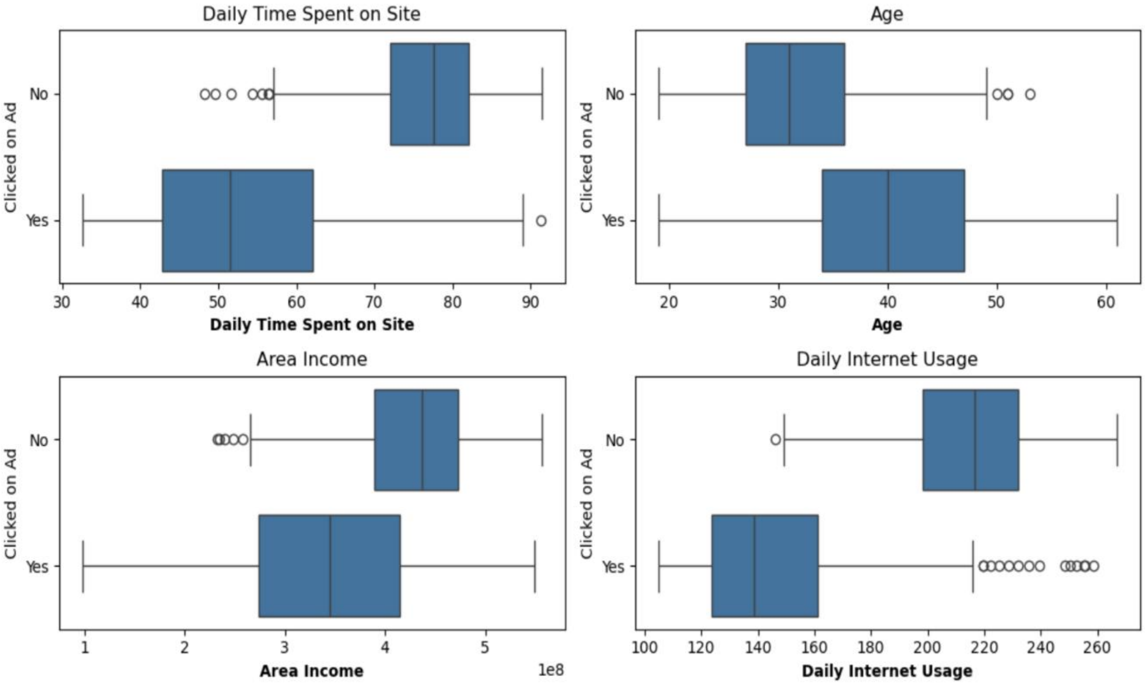


Fig 6.6 Boxplots of Numerical Features Between Users That Clicked and Didn't Click On Ad

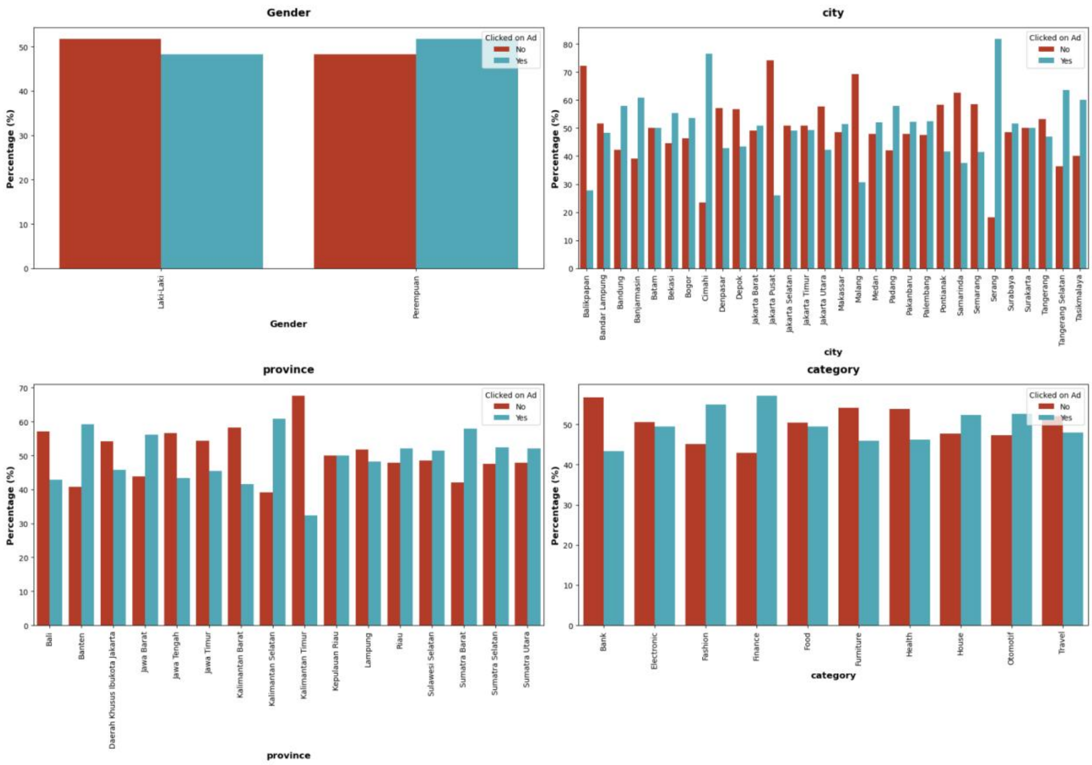


Fig 6.7 Bivariate Analysis - Barplot (gender, city, province, category)

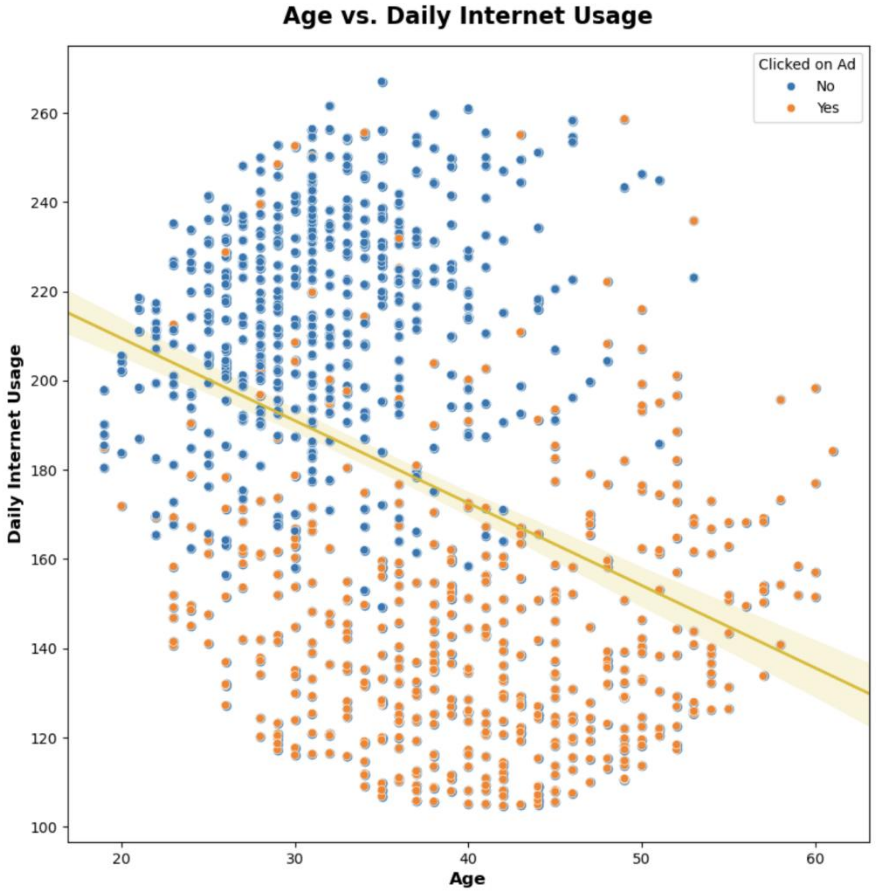


Fig 6.8 Age vs. Daily Internet Usage Scatterplot

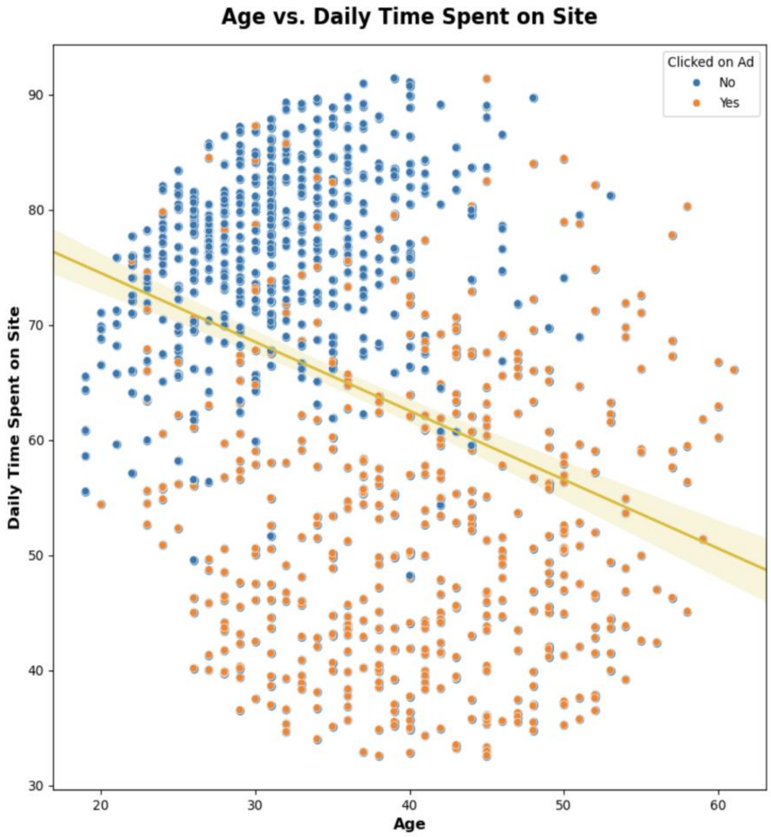


Fig 6.9 Age vs. Daily Time Spent on Site Scatterplot

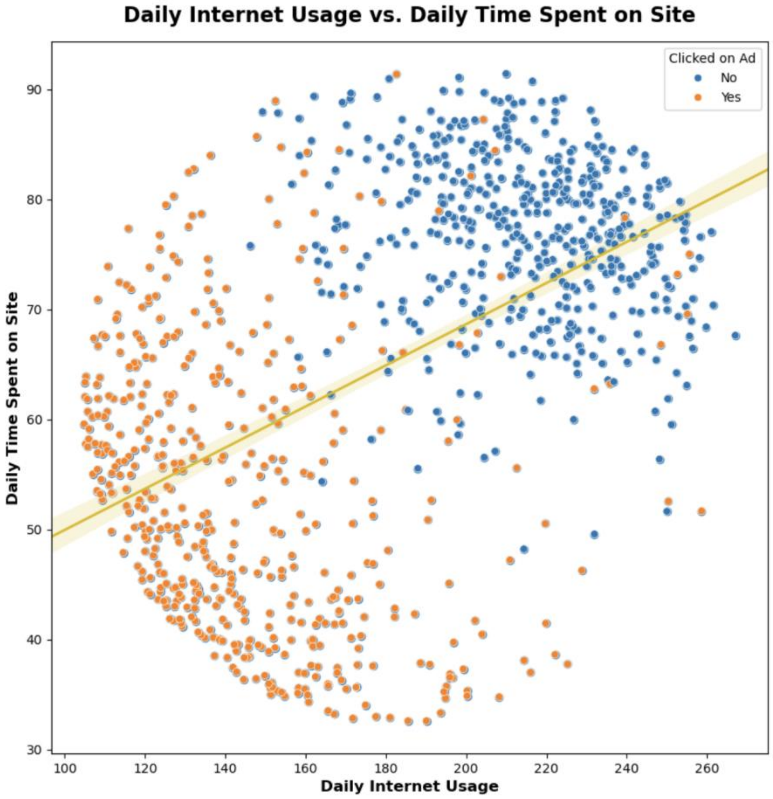


Fig 6.10 Daily Internet Usage vs. Daily Time Spent on Site Scatterplot

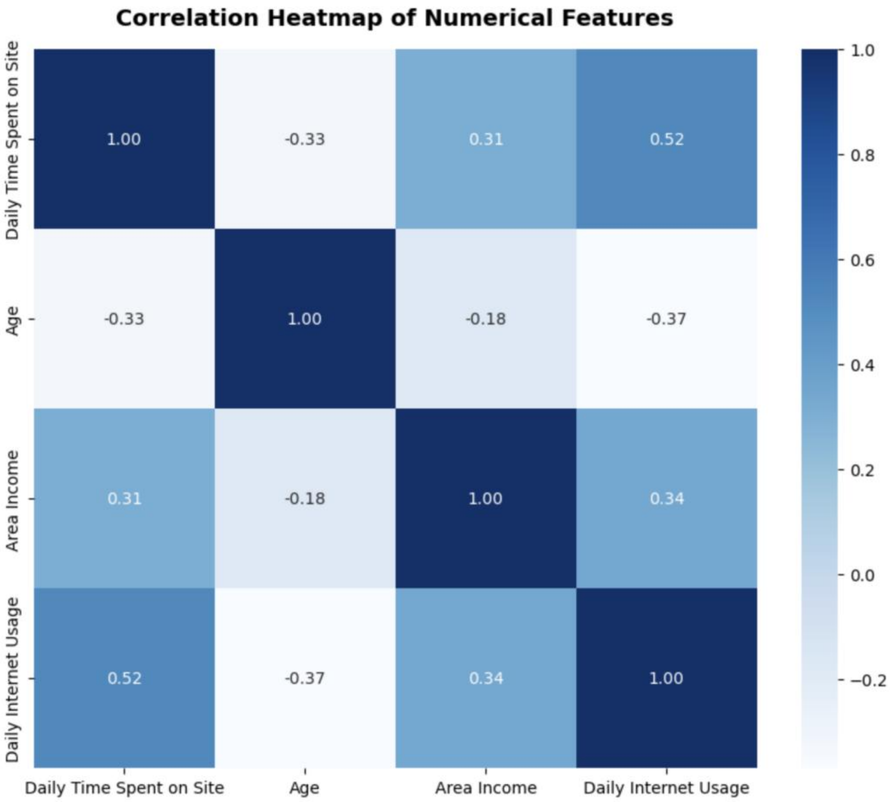


Fig 6.11 Correlation Heatmap of Numerical Features

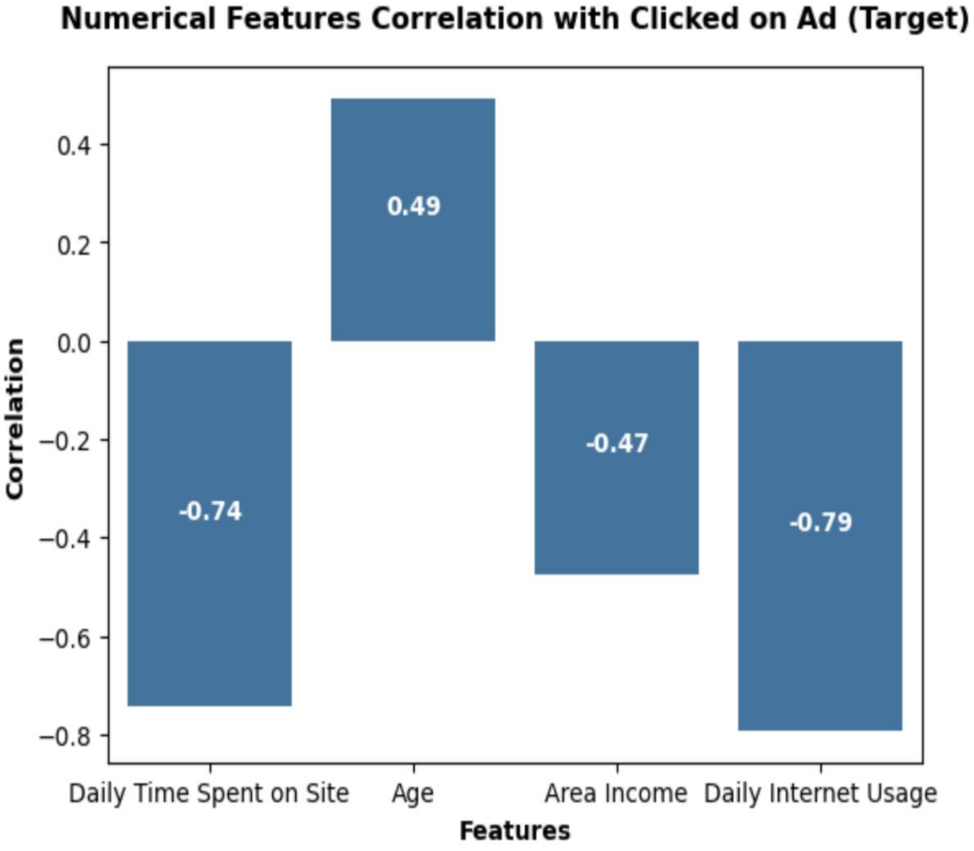


Fig 6.12 Correlation Bar Chart of Numerical Features and Target

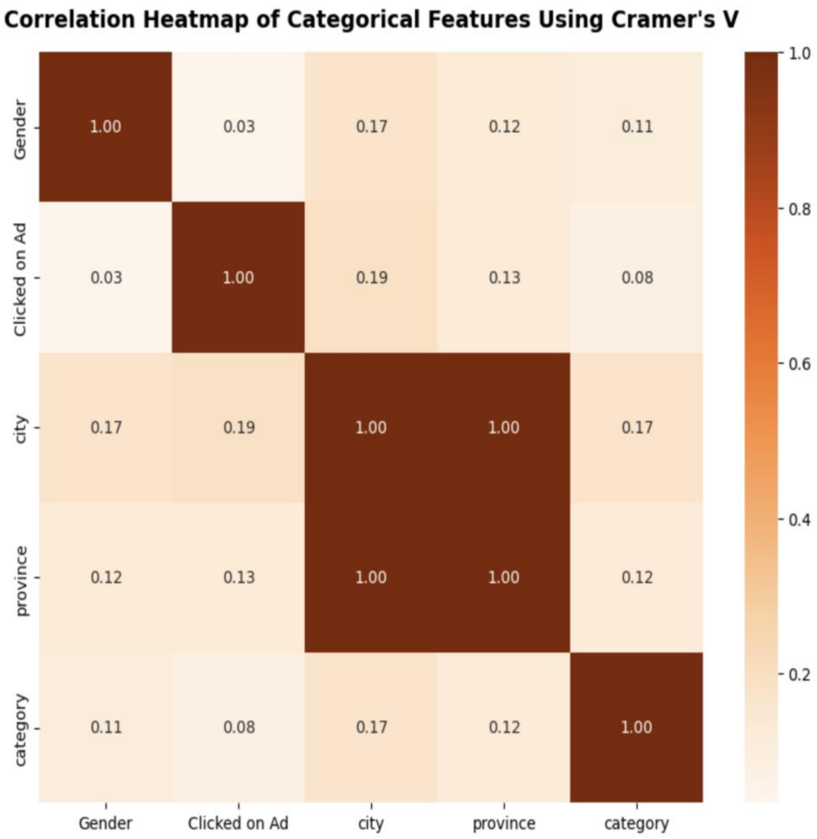
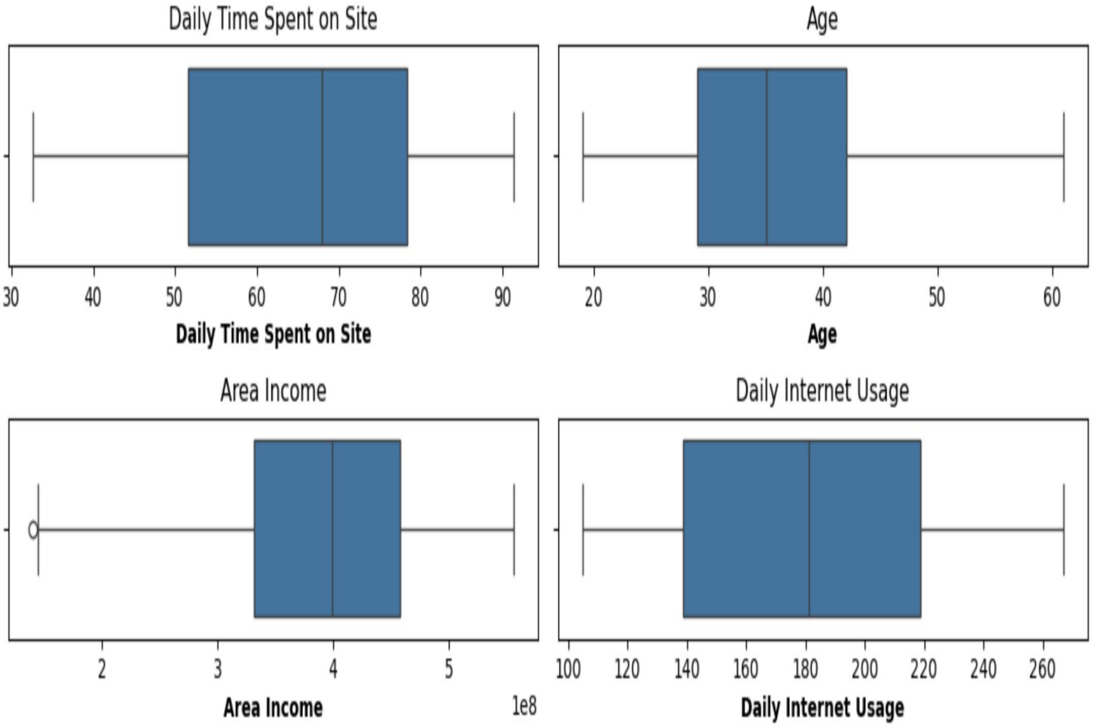


Fig 6.13 Correlation Heatmap of Categorical Features Using Cramer's V



As can be seen above, `Area Income` has no glaring outliers remaining.

Fig 6.14 Boxplot of Numerical before and after Outlier removal

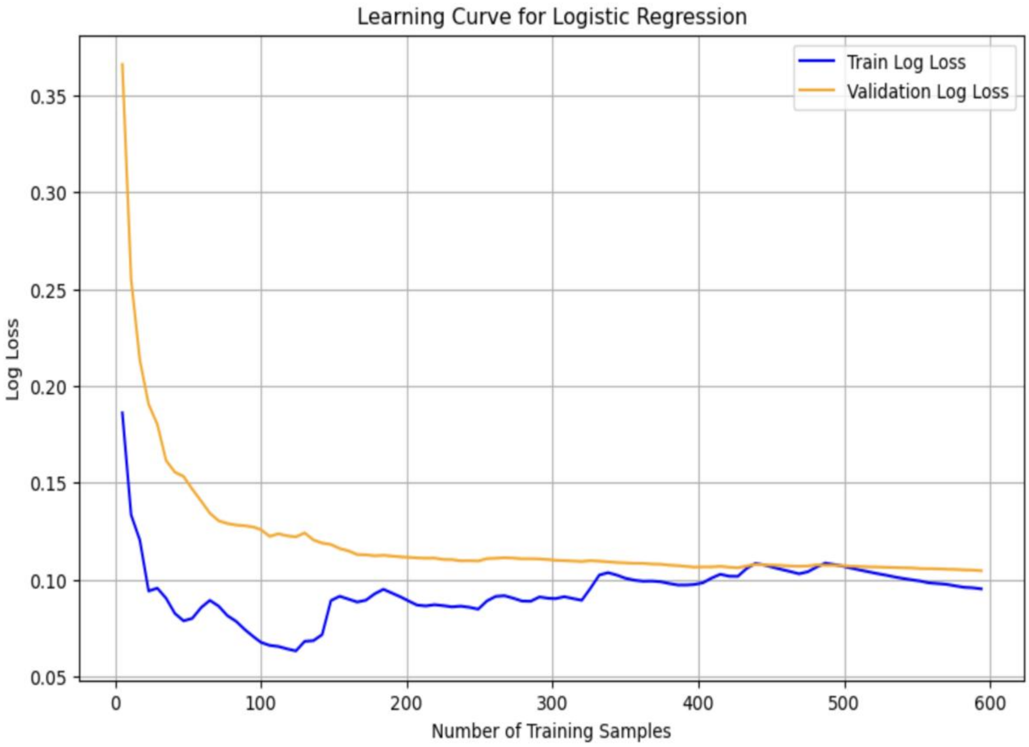


Fig 6.15 Learning Curve of Selected Model

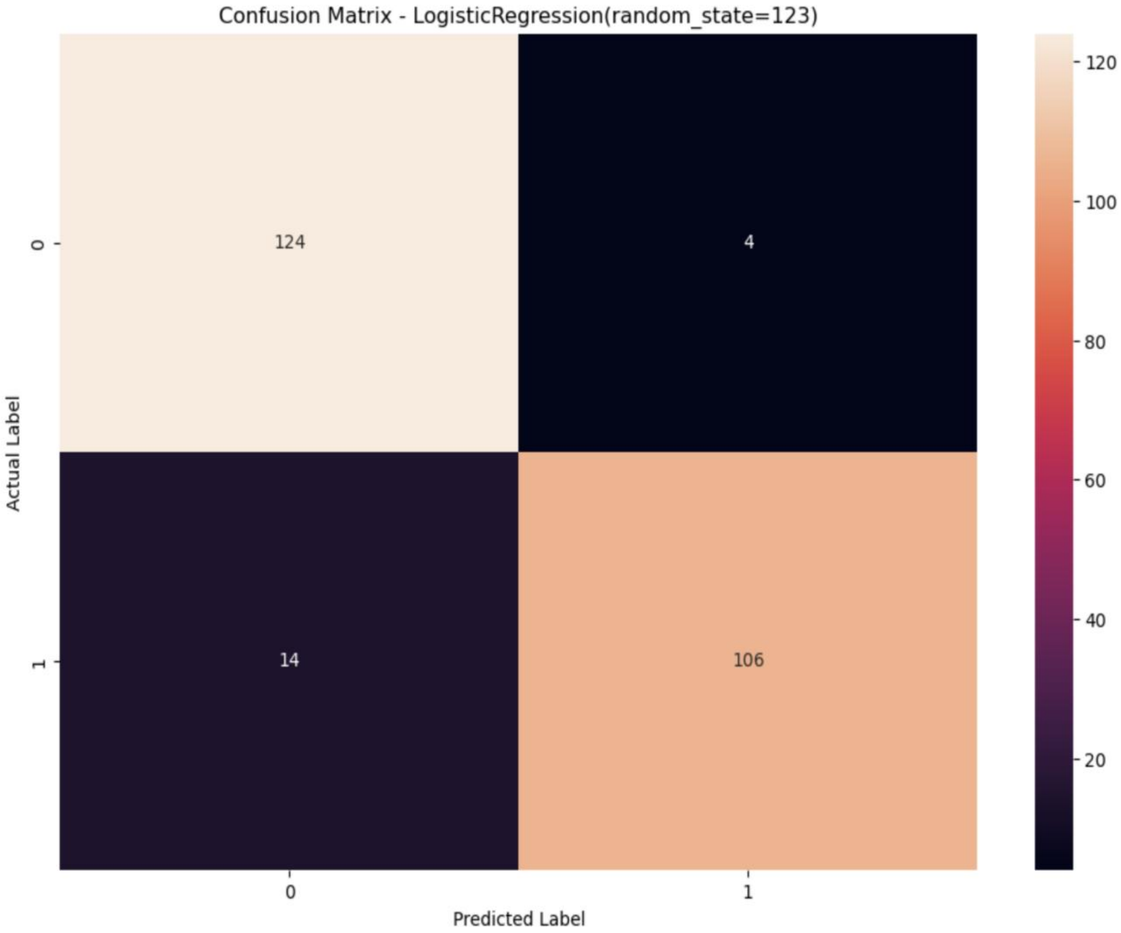


Fig 6.16 Confusion Matrix - Logistic Regression

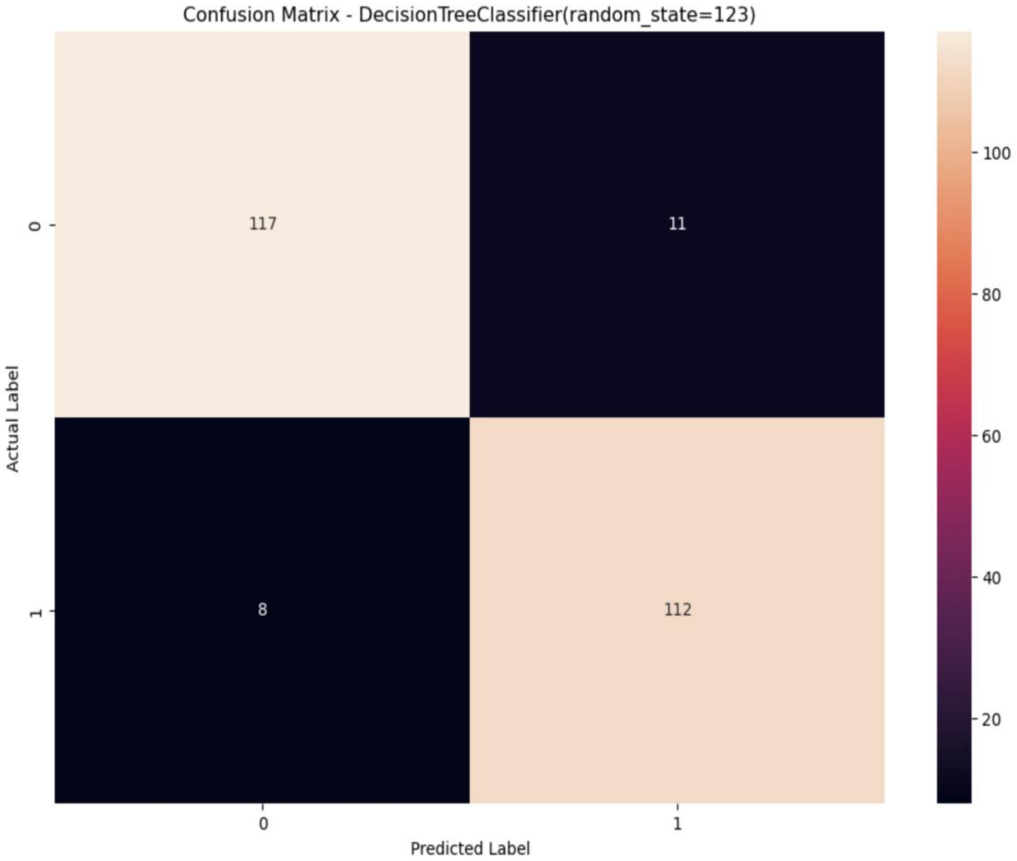


Fig 6.17 Confusion Matrix - Decision Tree Classifier

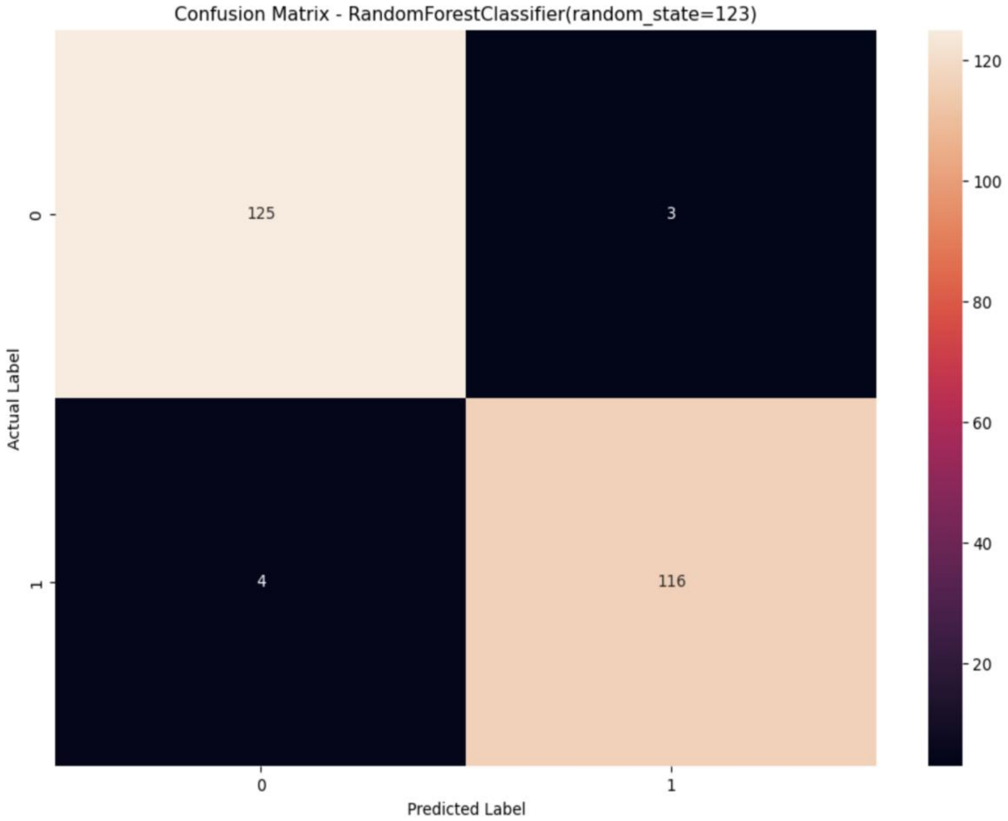


Fig 6.18 Confusion Matrix - Random Forest Classifier

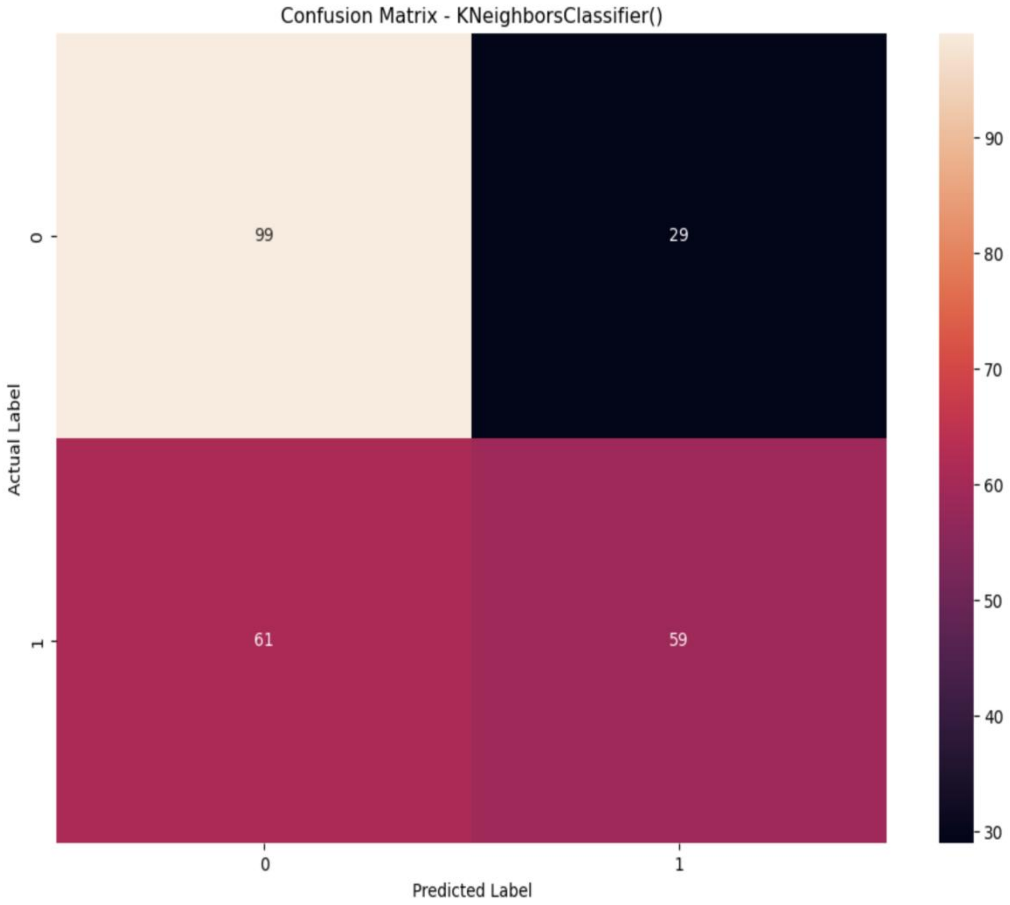


Fig 6.19 Confusion Matrix - KNeighbors Classifier

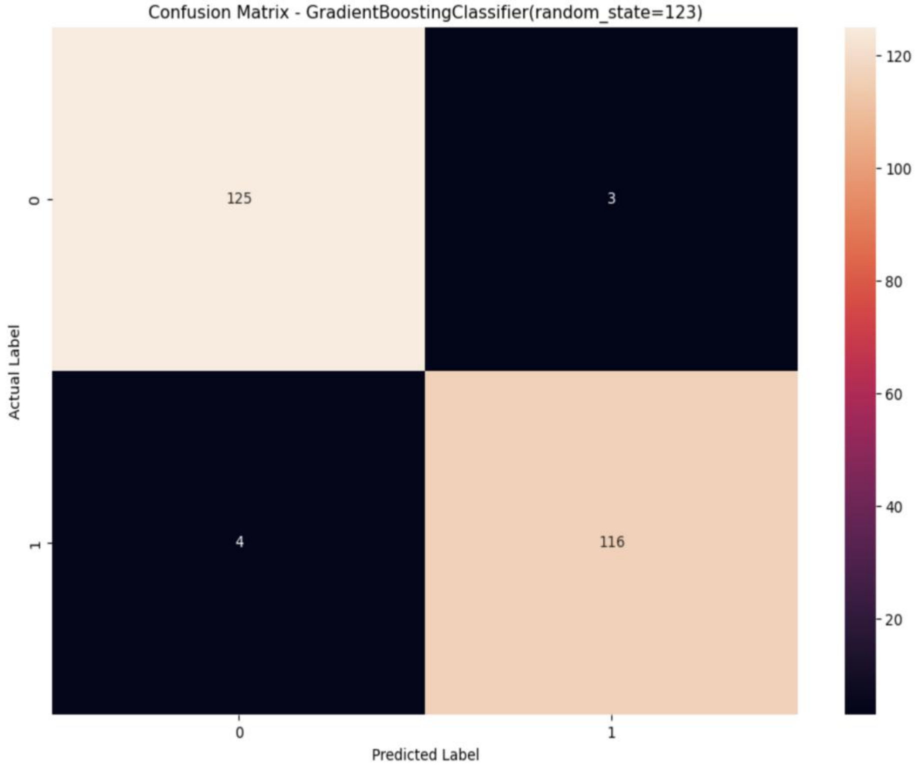


Fig 6.20 Confusion Matrix - Gradient Boosting Classifier

CHAPTER 7

CONCLUSION

This project demonstrates the practical application of machine learning in digital marketing by accurately predicting user ad-click behavior using key features such as age, daily internet usage, time spent on site, and income level. By implementing and comparing various models—including Logistic Regression, Decision Tree, Random Forest, KNN, Gradient Boosting, and XGBoost—the project identified high-performing algorithms that significantly improve ad targeting efficiency. Feature importance analysis revealed that users who spend less time on a site and have lower internet usage are more likely to click on ads, providing valuable insights for strategic ad placements and content personalization. As a result, the implementation led to a remarkable improvement in click-through rate from 50% to 96.8%, and an increase in advertising profit by 43.3%. These findings confirm the effectiveness of data-driven decision-making in enhancing user engagement, optimizing campaign performance, and maximizing returns on advertising investments. This system serves as a strong foundation for scalable, intelligent ad platforms in real-world applications.

REFERENCES

[1] Agarwal, D., & Chen, B. C. (2009). Predicting Clicks: Estimating the Click-through Rate for New Ads. Proceedings of the 2nd Workshop on Sponsored Search Auctions.

[2] McMahan, H. B., et al. (2013). Ad Click Prediction: A View from the Trenches. Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. DOI: [10.1145/2487575.2488200]

[3] Cheng, H.-T., et al. (2016). Wide & Deep Learning for Recommender Systems. Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, RecSys '16.

[4] Zhou, G., et al. (2018). Deep Interest Network for Click-Through Rate Prediction. Proceedings of the 24th ACM SIGKDD. arXiv:1706.06978

[5] Rendle, S. (2010). Factorization Machines. IEEE International Conference on Data Mining (ICDM). DOI: [10.1109/ICDM.2010.127]

[6] Juan, Y., et al. (2016). Field-aware Factorization Machines for CTR Prediction. Proceedings of the 10th ACM Conference on Recommender Systems.

[7] Zhang, R., et al. (2014). Learning to Rank with Deep Neural Networks for CTR Prediction. arXiv:1404.3360

[8] Yang, Y., et al. (2019). Operation-aware Neural Networks for User Response Prediction. arXiv:1904.12579

[9] Friedman, J. (1999). Greedy Function Approximation: A Gradient Boosting Machine. Technical Report, Dept. of Statistics, Stanford University.

[10] Hastie, T., Tibshirani, R., & Friedman, J. (2001). The Elements of Statistical Learning. Springer-Verlag.

[11] Agarwal, D., & Chen, B. C. (2010). Bayesian Click-Through Rate Prediction for Sponsored Search. Proceedings of the VLDB.

[12] Zhang, W., et al. (2014). Optimal Real-Time Bidding for Display Advertising. Proceedings of the 20th ACM SIGKDD. DOI: [10.1145/2623330.2623633]

[13] Did-it, Enquiro, & Eyetools. Eye Tracking Study. [http://www.enquiro.com/eye-tracking-pr.asp]

[14] Bartz, K., Murthi, V., & Sebastian, S. (2006). Logistic Regression and Collaborative Filtering for Sponsored Search Term Recommendation. Proceedings of the 2nd Workshop on Sponsored Search Auctions.

[15] Nocedal, J., & Wright, S. J. (1999). Numerical Optimization. Springer-Verlag.