Ad Click-Through Rate Prediction for Digital Marketing

Presented by W. Aalia Fathima

INTRODUCTION

In today's digital world, online advertising is a primary way for businesses to reach potential customers. However, showing irrelevant ads wastes advertising budgets and leads to poor user engagement.

Click-Through Rate (CTR) represents the likelihood of a user clicking on an advertisement, making it a critical metric in digital marketing.

This project uses historical user and ad data with machine learning models to predict CTR, demonstrating how data-driven approaches can improve online advertising efficiency.

Problem Statement

In digital marketing, businesses invest heavily in online advertising to reach customers and drive engagement. However, existing ad-serving systems often use simplistic, rule-based targeting that fails to capture complex user behavior and ad context.

This project addresses the need for a machine learning-based CTR Prediction Model to accurately predict the likelihood of a user clicking on an advertisement by analyzing historical user behavior, demographic data, and ad-specific features.

By solving this problem, the system helps:

- Enable data-driven decisions in digital advertising.

Objective

The primary objective of this project is to develop a machine learning-based CTR Prediction Model to improve the accuracy and efficiency of online advertising.

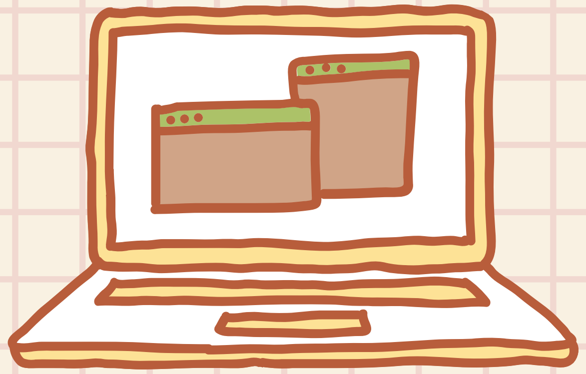
- Predict the likelihood of a user clicking on an advertisement using historical user behavior, demographic data, and ad-specific features.
- Improve user experience by displaying relevant, personalized advertisements.
- Build and deploy a user-friendly Streamlit application for real-time CTR predictions.
- Enhance practical skills in data science, machine learning, and deployment within a real-world digital marketing context.

Dataset

	Daily Time Spent on Site	Age	Area Income	Daily Internet Usage	Ad Topic Line	City	Gender	Country	Timestamp	Clicked on Ad
0	62.26	32.0	69481.85	172.83	Decentralized real-time circuit	Lisafort	Male	Svalbard & Jan Mayen Islands	2016-06-09 21:43:05	0
1	41.73	31.0	61840.26	207.17	Optional full-range projection	West Angelabury	Male	Singapore	2016-01-16 17:56:05	0
2	44.40	30.0	57877.15	172.83	Total 5thgeneration standardization	Reyesfurt	Female	Guadeloupe	2016-06-29 10:50:45	0
3	59.88	28.0	56180.93	207.17	Balanced empowering success	New Michael	Female	Zambia	2016-06-21 14:32:32	0
4	49.21	30.0	54324.73	201.58	Total 5thgeneration standardization	West Richard	Female	Qatar	2016-07-21 10:54:35	1

Dataset Usage:

- Used for training and evaluating multiple machine learning models.
- Features engineered to extract patterns influencing ad click behavior.
- Data preprocessed for handling missing values, encoding categorical features, and ensuring consistency for model training and deployment.



Tools and Technologies Used

Platforms:

- Kaggle Notebook: Used for data exploration, preprocessing, model training, and evaluation in a cloud-based environment.
- VS Code + Streamlit: Used to develop and test the Streamlit app locally for real-time CTR prediction.

Libraries and Frameworks:

- pandas, numpy: Data loading, manipulation, and numerical computations.
- matplotlib, seaborn: Data visualization during EDA for understanding feature distributions and relationships.
- scikit-learn: Model implementation (Logistic Regression, Decision Tree, Random Forest), preprocessing utilities, and evaluation metrics.

Tools and Technologies Used

Libraries and Frameworks:

- XGBoost: Implemented the final prediction model with high accuracy and efficiency.
- joblib: Model serialization for saving and loading the trained model seamlessly in the app.
- Streamlit: For building a user-friendly, interactive web app for real-time CTR predictions.
- datetime: For extracting and managing time-based features (day, month, hour) from the dataset.

Methodology / Workflow

Data Collection

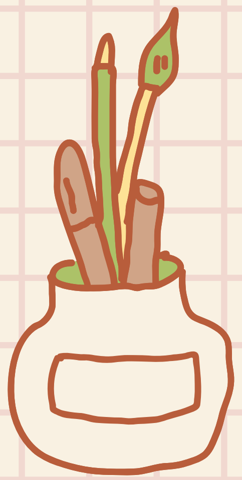
- Sourced CTR dataset from Kaggle.

Exploratory Data Analysis(EDA)

- Analyzed feature distributions and patterns.
- Visualized correlations to understand influencing factors.
- Detected and handled outliers.

Data Preprocessing

- Encoded categorical features (Label & Frequency Encoding).
- Extracted time-based features (day, month, hour).



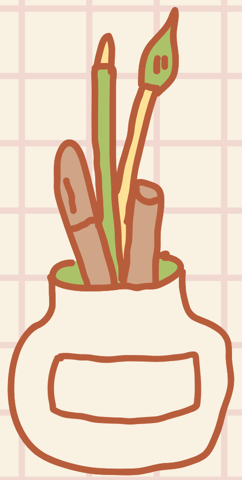
Methodology / Workflow

Model Training & Evaluation

- Trained:
  - Logistic Regression
  - Decision Tree
  - Random Forest
  - XGBoost (final model with best accuracy)
- Evaluated using accuracy, MAE, and RMSE.

Model Saving and Deployment

- Saved the trained XGBoost model and encoders using Joblib.
- Built a Streamlit app to:
- Take real-time user inputs.
- Predict CTR instantly.
- Display click probability and feature importance insights.



System Architecture

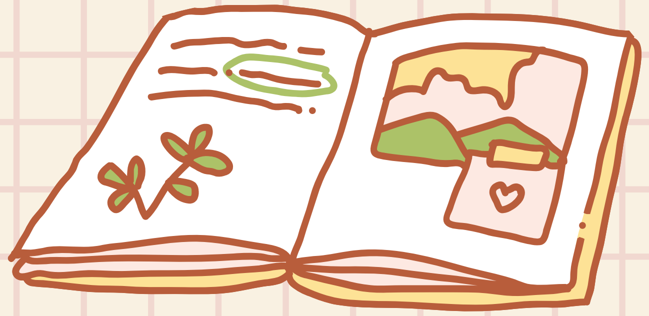
1.Data Layer

- Historical Data: Loads ad click data from Kaggle using pandas for training.
- Real-Time Input: Uses Streamlit forms to collect user data for live predictions.

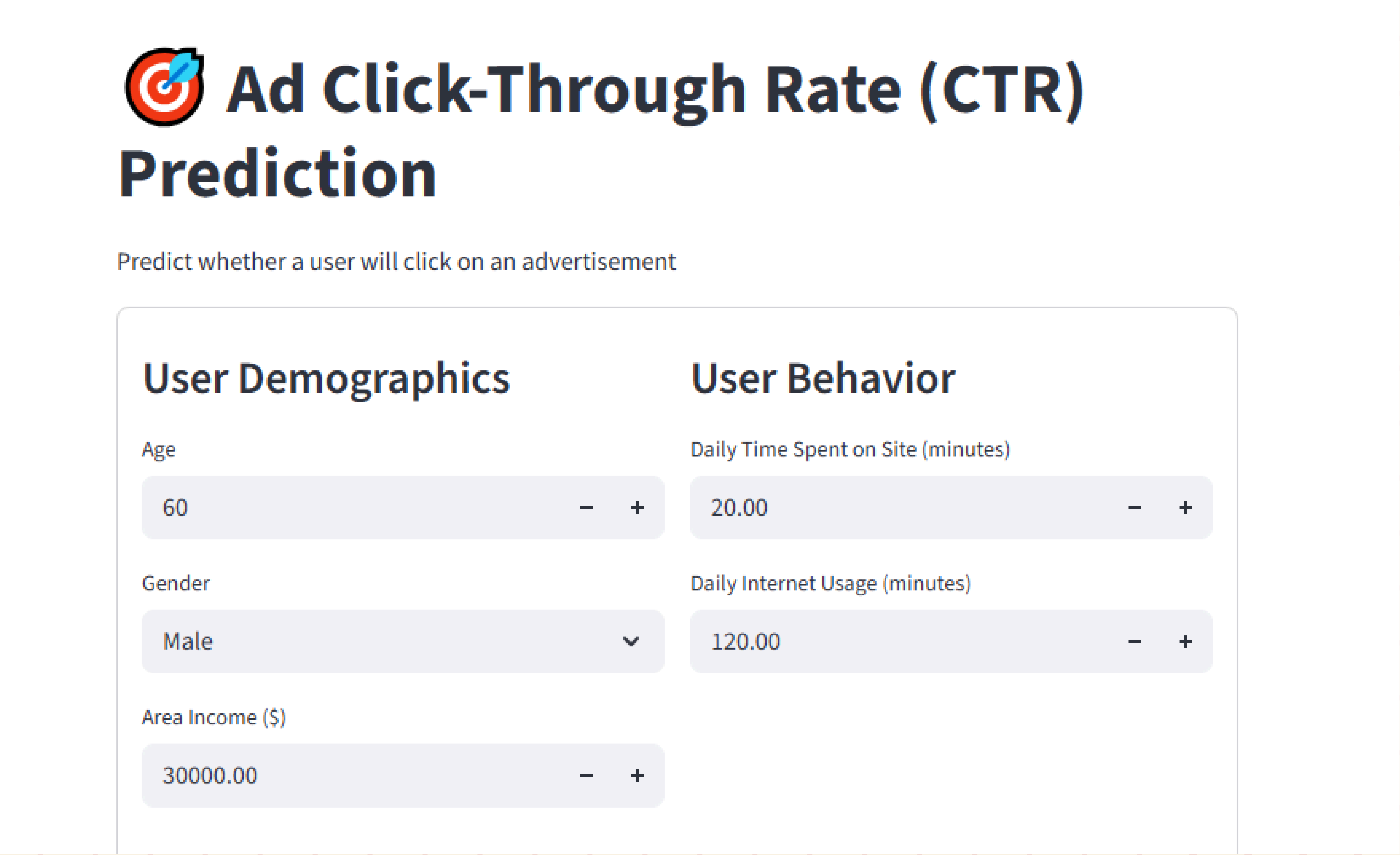
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 10 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   DailyTime_Spent_on_Site   10000 non-null  float64
 1   Age                       10000 non-null  float64
 2   Area_Income               10000 non-null  float64
 3   Daily_Internet_Usage      10000 non-null  float64
 4   Ad_Topic_Line             10000 non-null  object
 5   City                      10000 non-null  object
 6   Gender                    10000 non-null  object
 7   Country                   10000 non-null  object
 8   Timestamp                 10000 non-null  datetime64[ns]
 9   Clicked_on_Ad             10000 non-null  int64
dtypes: datetime64[ns](1), float64(4), int64(1), object(4)
memory usage: 781.4+ KB
```

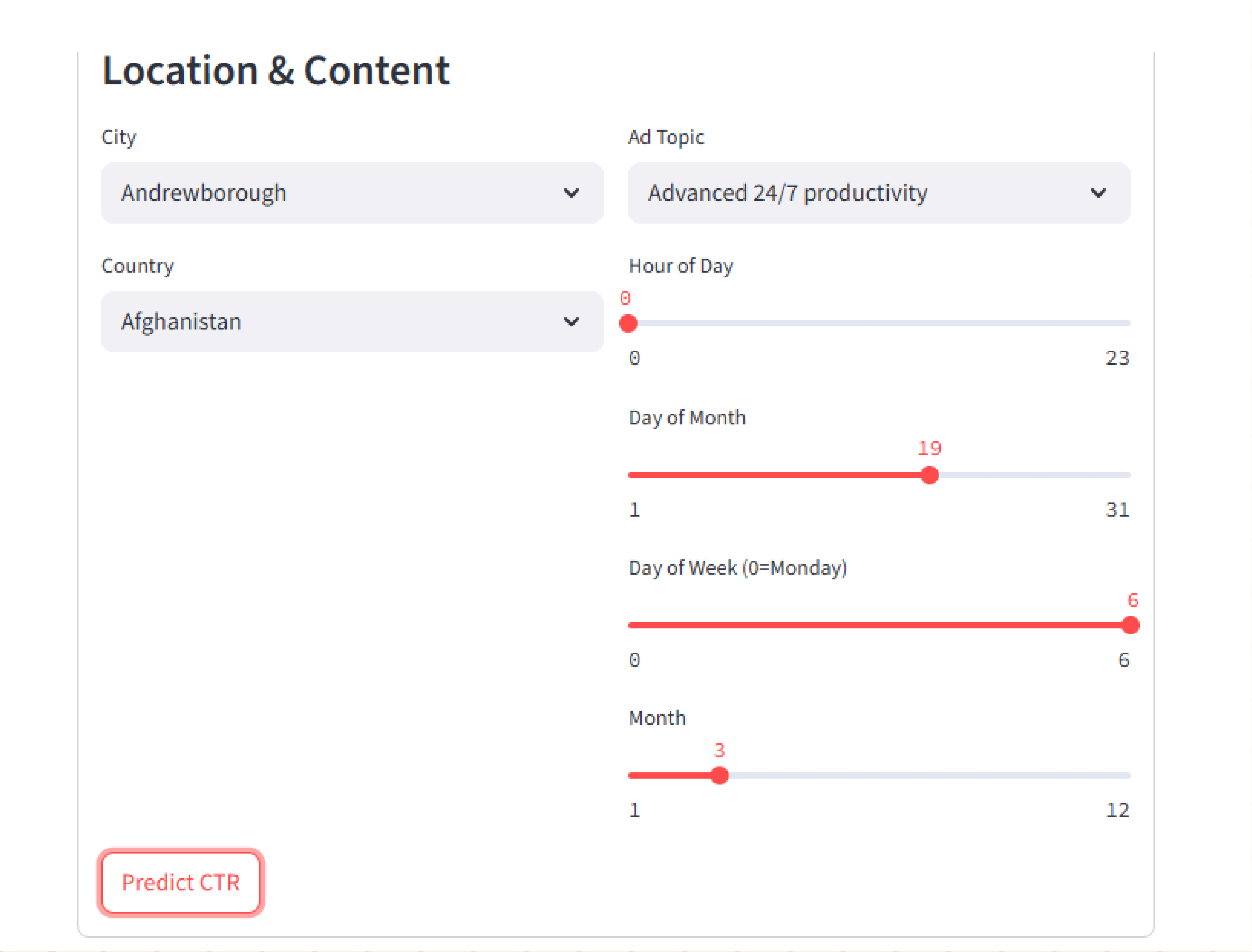
```
data.nunique()
```

```
DailyTime_Spent_on_Site    460
Age                         39
Area_Income                524
Daily_Internet_Usage       505
Ad_Topic_Line              559
City                       521
Gender                       2
Country                    207
Timestamp                  567
Clicked_on_Ad                2
dtype: int64
```



System Architecture



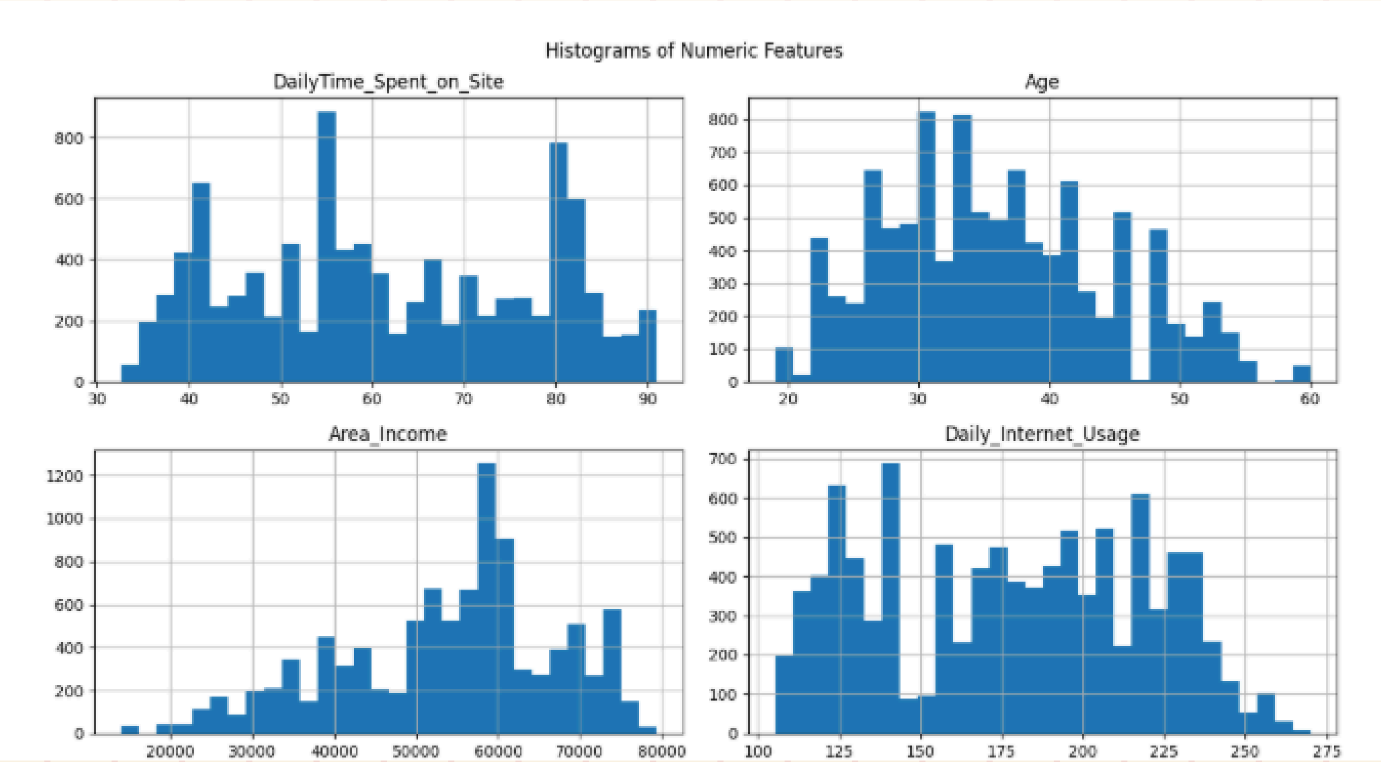


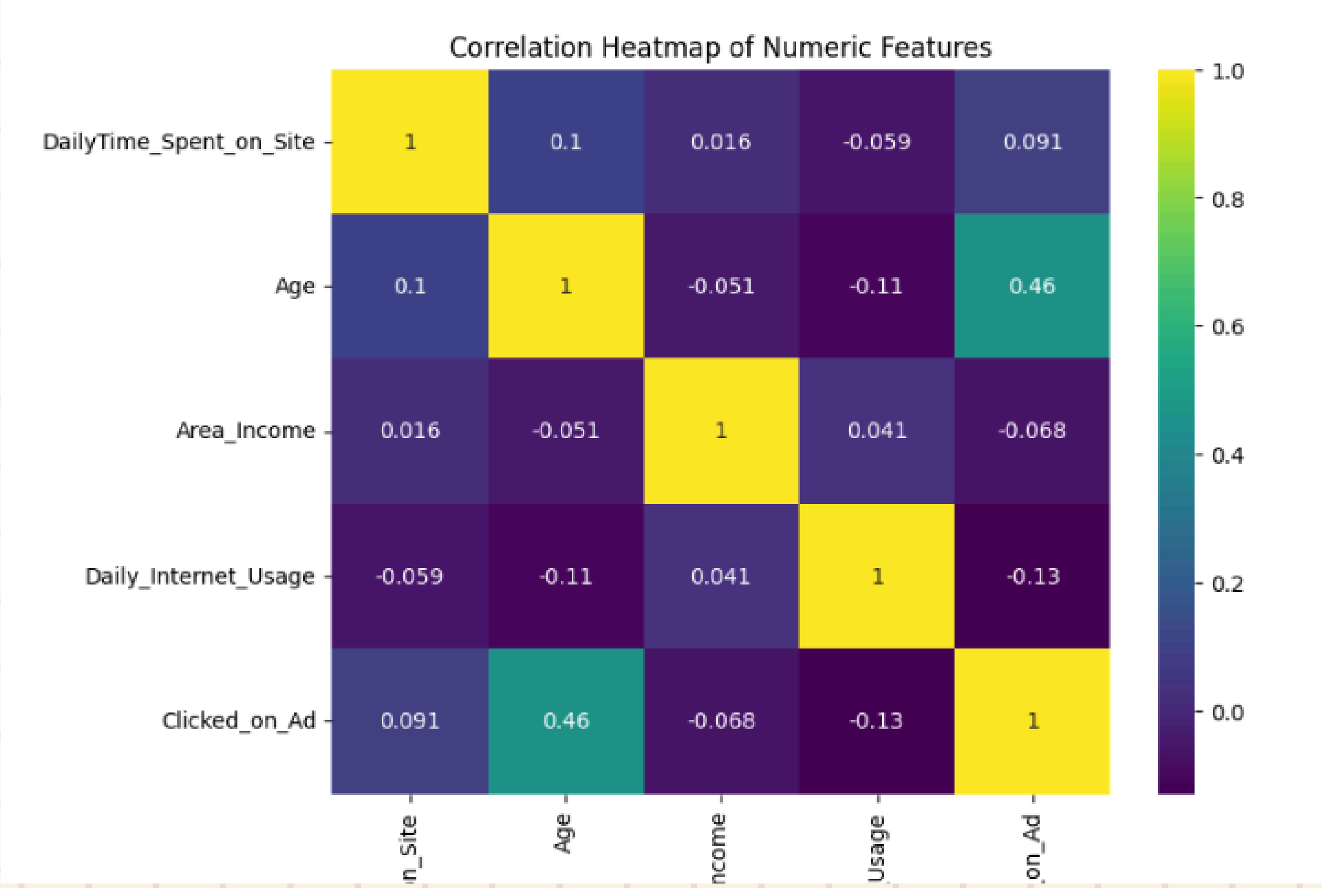
System Architecture

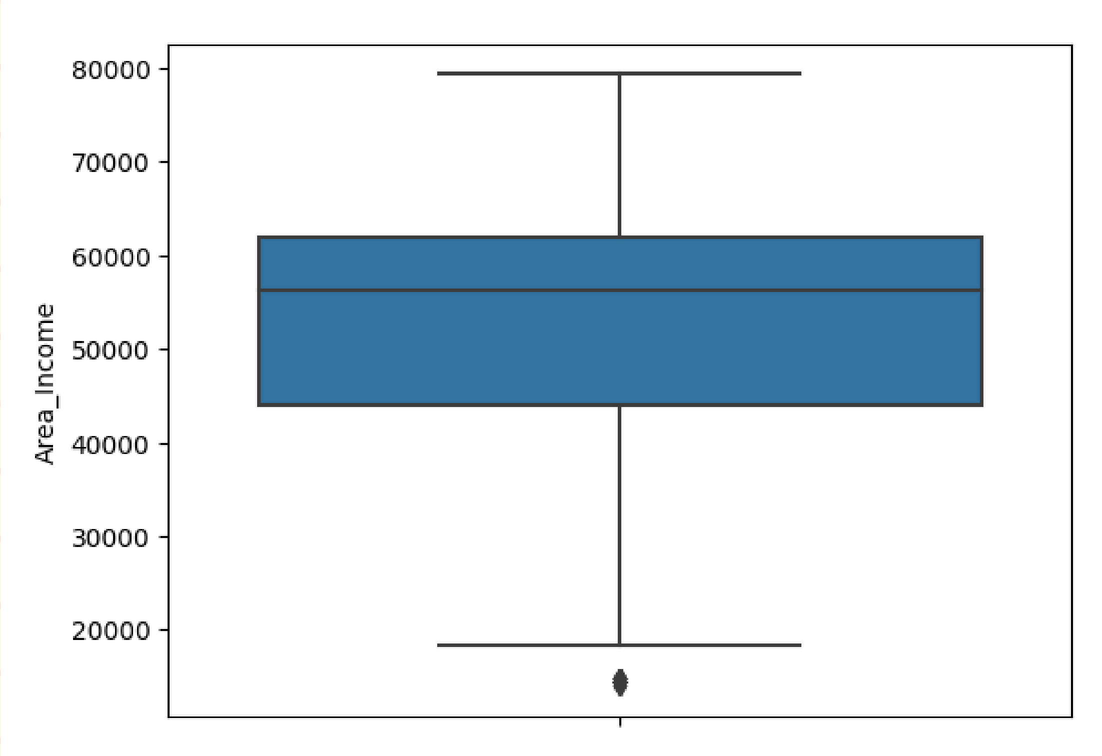
2. Analysis & Processing Layer

EDA Steps

- Descriptive Statistics
- Visualization
- Outlier Detection







System Architecture

2. Analysis & Processing Layer

preprocesses data

- Encoding categorical features
- Feature Engineering

```
Out[19]:
0    1
1    1
2    0
3    0
4    0
Name: Gender, dtype: int64
```

day_of_month	hour_of_day	day_of_week	month
9	21	3	6
16	17	5	1
29	10	2	6
21	14	1	6
21	10	3	7

City_frequency	City_encoded	Country_frequency	Country_encoded	Ad_Topic_encoded
261	234	6	173	96
109	460	130	165	301
205	379	9	71	484
110	269	39	204	24
16	495	223	148	484

System Architecture

3. Model Layer

Model Training:

- Logistic Regression: A simple linear model that predicts the probability of a user clicking on an ad.
- Decision Tree: Splits data into branches using rules to capture non-linear patterns.
- Random Forest: Combines multiple decision trees to improve accuracy and reduce overfitting.
- XGBoost: An advanced boosting algorithm that builds trees sequentially for high accuracy (selected for deployment).

System Architecture

3. Model Layer

Model Evaluation:

- Evaluated using:
  - Accuracy
  - Mean Absolute Error (MAE)
  - Root Mean Square Error (RMSE)
- XGBoost achieved highest accuracy, making it the final choice.

Model Saving:

- Uses Joblib to save the trained XGBoost model and preprocessing artifacts.

Model Loading:

- Loads the saved model in the Streamlit app for real-time CTR predictions.

System Architecture

4. Application Layer

- User Interface: Built using Streamlit for a clean, interactive web app.
- Real-Time User Inputs
- Prediction Display:
  - Uses the loaded XGBoost model to predict CTR instantly.
  - Shows:
    - Click / No Click prediction
    - Click probability
    - Feature importance for transparency
    - Targeting insights (Excellent, Moderate, Low)

System Architecture

4. Application Layer

❌ WON'T CLICK

Click Probability

31.1%

📊 Insights

❌ **Poor targeting.** This user profile is unlikely to engage.

🎯 WILL CLICK

Click Probability

80.6%

📊 Insights

🎯 **Excellent targeting!** This user profile shows high engagement potential.

🎯 WILL CLICK

Click Probability

62.8%

📊 Insights

⚠️ **Moderate targeting.** Consider optimizing ad content or timing.

RESULT

The CTR Prediction Model successfully predicts user ad clicks, improving targeting and campaign efficiency.

DEPLOYED LIVE:

Access the deployed application here:
https://ctr-prediction-ds.onrender.com

Future Scope

1. **Integration of Deep Learning**
   - Explore Deep Neural Networks (DNN) for capturing complex, high-order feature interactions.
   - Potential for even higher accuracy on large datasets.
2. **Real-Time Data Streaming**
   - Use Kafka or similar tools to enable continuous model updates and live predictions as user data flows in.
3. **Automated Model Retraining**
   - Build pipelines for scheduled or event-triggered retraining to maintain prediction accuracy as user behavior evolves.

CONCLUSION

In conclusion, the CTR Prediction Model using machine learning demonstrates how data-driven approaches can improve digital advertising by predicting the likelihood of a user clicking on an ad. By leveraging user behavior, demographics, and ad features, the project helps advertisers reduce budget wastage and improve user engagement. Using a systematic workflow, the model was trained and deployed in a user-friendly Streamlit app for real-time predictions. This project showcases the practical application of machine learning in solving real-world problems, bridging the gap between theoretical learning and industry needs.

THANKYOU