University: Emory Goizueta Business School

Program: MSBA class 2024

Title: AI & ML Group Final Project

Team members:

- Maliki Uwase
- Sunday David
- Basani Reddy Siva Prasad Naidu
- Yaeeun Lee

Part 1: Introduction

Project Overview:

The landscape of digital advertising has been rapidly evolving with the integration of Artificial Intelligence and Machine Learning technologies. Our project, situated at the forefront of this evolution, aims to harness these advancements by focusing on predicting click-through rates for online advertisements. This endeavor is not only a pivotal aspect of digital marketing strategies but also a complex challenge in the field of predictive analytics.

The core objective of our project is to develop a robust predictive model capable of accurately forecasting the likelihood of a user clicking on an advertisement. This task involves analyzing a substantial dataset containing over 30 million records, spanning nine days from October 21, 2014, to October 29, 2014. The dataset comprises various features related to the advertisements, such as ad identifiers, site categories, app domains, device types, and several anonymized categorical variables.

Our approach is to apply Machine Learning algorithms to this rich dataset to uncover underlying patterns and relationships that influence ad click behavior. The significance of this project lies in its potential to revolutionize how businesses target and engage with their audience in the digital space, making advertising efforts more efficient and cost-effective.

In accomplishing this, we not only aim to enhance the precision of ad targeting but also contribute to the broader understanding of user engagement in the digital advertising realm. This project stands as an exemplar of the practical application of AI & ML techniques in solving

real-world business challenges and is a testament to the transformative power of these technologies in the ever-evolving digital landscape.

Part 2: Team Introduction:

1. **Data Preprocessing and Feature Engineering (Maliki Uwase)**:
    - Cleans and preprocesses the dataset.
    - Handles feature selection and engineering.
2. **Model Development and Training Lead (Sunday David)** :
    - Selects and applies machine learning algorithms.
    - Oversees model development and training.
3. **Evaluation and Analysis Expert (Siva)**:
    - Implements and manages evaluation metrics.
    - Analyzes results and model comparisons.
4. **Project Management and Coordination Lead (Yaean Lee)**:
    - Coordinates project activities and timelines.
    - Manages communication and collaboration among team members.

Shared Responsibility - Documentation and Reporting:

- All team members contributed to the documentation and reporting process.
- Each member documents their respective sections and collaborates on the final report to ensure consistency and completeness.

This approach ensures that while each team member has a specific focus area, the crucial task of documentation and reporting is a collaborative effort, reflecting our collective contributions and insights.

Part 3: Project Description

- **Problem Statement**:

  The primary challenge our project addresses is the prediction of click-through rates for online advertisements. This involves determining the probability that a user will click on an ad based on various features associated with the ad. Accurately predicting ad clicks is crucial for optimizing digital advertising strategies and enhancing the efficiency of online marketing campaigns.

- **Dataset Description**:

  Our analysis utilizes the dataset provided in "ProjectTrainingData.csv," comprising over 30 million records spanning from October 21, 2014, to October 29, 2014. This dataset encompasses a range of variables, including:

- ☐ **Ad Identifiers**: Such as 'id', 'site_id', 'app_id', and others that uniquely define the ad's context.
- ☐ **User Interaction**: Represented by the 'click' variable, indicating whether the ad was clicked (1) or not (0).
- ☐ **Temporal Data**: The 'hour' variable, showing the date and hour of ad display.
- ☐ **Categorical Variables**: Including anonymized variables like C1, banner_pos, site_category, etc.
- ☐ **Device Information**: Like 'device_id', 'device_type', and 'device_conn_type'.

Preprocessing steps involved cleaning the data to handle missing or inconsistent entries, encoding categorical variables, and normalizing the features for effective model training.

- **Goals and Objectives**:

  Our project's primary aim is to create a machine learning model that accurately predicts online ad click-through rates. Key objectives include:

  - **Data Understanding :**Identify the need for data cleaning, preprocessing, and feature engineering.
  - **Feature Selection** : Gleaning insights to understand factors influencing ad clicks.
  - **Model Building and Optimization**: Building and fine-tuning a model for reliable prediction of ad clicks.
  - **Real-World Application**: Developing a practical tool for enhancing digital advertising strategies.

  Through these goals, we intend to make a substantial impact in the digital advertising domain, leveraging AI and ML for strategic advancements.

Part 4: Methodology

☐ Exploratory Data Analysis (EDA)

Our EDA focused on understanding the features through summary statistics, data type analysis, and distribution examination. Key findings include:

- **Data Completeness**: The dataset contained no missing values.

- **Data Types**: Features were a mix of object and integer data types.
- **Class Imbalance**: Notable imbalance in the binary target variable **[Figure 1]**

- **Label Encoding**: Object data type features were label-encoded for consistency across train and test datasets.
- **Feature Derivation**: From the 'hour' column, we derived additional features like the day of the week and time of day (morning, afternoon, etc.).
- **Correlation Analysis**: Post-encoding, a correlation matrix was generated to assess relationships between features **[Figure 2]**.

- **Data Sampling**: For manageability, we sampled 5,000,000 records from the training dataset.

☐ **Feature Engineering**

Feature selection was performed using Random Forest to identify features with high predictive power. And I ended up selecting top **15** predictor features This step was pivotal in enhancing model accuracy. A graph visualizing the feature importance ranking is appendix **[Figure 3]**:

☐ **Model Selection and Baseline Models**

Initial modeling involved creating baseline models with 1,000,000 records using Decision Trees and Logistic Regression. We evaluated these models using:

- **Log Loss** as the primary performance metric.
- Other metrics like **Accuracy and F1 Score**. Results of these baseline models will be presented in a table below:

Model	Decision Tree	Random Forest	Logistic regression
Log loss	**8.91**	**0.58**	**0.45**

F1 Score	0.28	0.22	0.00
Accuracy	0.75	0.81	0.83

Furthermore, while our current progress points towards the efficacy of ensemble methods, we also remain open to exploring neural networks. The potential of neural nets, especially in handling large-scale and complex datasets, makes them a worthwhile consideration for further enhancing our model's performance.

- ☐ **Modeling part 2 with hyperparameter tuning for imbalance class**
  - Random Forest model:

    In our efforts to improve the Random Forest model for predicting ad click-through rates, we focused on tuning `n_estimators` (number of trees) and `max_depth` (depth of each tree). We tested `n_estimators` at 100, 200, and 300, and `max_depth` at 10, 15, and 20, aiming to optimize model complexity and accuracy.

    We split the data into a 70:30 training-validation ratio and employed grid search to evaluate different RandomForestClassifier configurations, using log loss as our metric. The best performance was achieved with `n_estimators` at 300 and `max_depth` at 20, resulting in a log loss of **0.39762833751658744**. This configuration significantly improved prediction accuracy compared to our baseline models, demonstrating a refined model adept at discerning complex patterns in the data. **[Figure 4,5]**

  - Neural Networks model:
    The neural network, trained over 15 epochs, exhibited a substantial improvement in performance, as evidenced by a log loss that reduced to 0.4556. This reduction in log loss, from initially high values to a much lower figure, indicates a significant enhancement in the model's predictive accuracy, achieving an approximate accuracy of 83%. The log loss of 0.4556 is particularly indicative of the model's efficiency in classifying the dataset with a high degree of reliability.

  - XGBoost model:

The XGBoost classifier, tailored for binary classification, was trained on a dataset. Key parameters included 100 estimators, a max depth of 20, and a learning rate of 0.1. Post-training, the model's predictive performance on the test set was evaluated using a custom log loss function, resulting in a log loss of **0.3993.** This value indicates a strong predictive accuracy, reflecting the model's effectiveness in distinguishing between the binary classes. The relatively low log loss underscores the model's reliability in making probabilistic predictions, which is a critical aspect of binary classification tasks. This is the best result we got, after manually tuning parameters. **[*Figure 6*]**

☐ **Modeling part 3 with hyperparameter tuning without imbalance class**

Overview:

To tackle the class imbalance in our dataset, we used undersampling, a crucial step in preparing data for modeling. Our original data had a significant skew, with more 0's than 1's. By undersampling, we kept all the 1's and reduced the 0's to balance the classes, aiming for a total of 10 million samples. This is vital because it helps our model learn from both classes more effectively, enhancing its accuracy, especially for the minority class. Additionally, we focused on the most influential variables, selecting the top 15 based on feature importance. This method hones the model's attention on the most impactful factors, fostering better predictions and efficiency. Combining undersampling with selective feature inclusion allows us to build a more robust and reliable model.

- Neural Networks model:

  After addressing the imbalance in our dataset, we developed a neural network model using Keras. This Sequential model, designed for binary classification, consisted of four layers, configured with an Adam optimizer (learning rate 0.001) and binary crossentropy loss. We split the data into an 80% training and 20% validation set. The training was carried out over 10 epochs with a batch size of 64, and model performance was closely monitored with a checkpoint callback. Post-training evaluations, including log loss, accuracy, F1 score, and AUC, were conducted to assess the model’s efficacy in predicting online ad click-through rates. Log loss was 0.5432.

- XGBoost model:

  In advancing our project, we also re-trained an XGBoost classifier, configured with parameters like 100 estimators, a maximum depth of 20, and a learning rate of 0.1, focusing on binary classification. The model was trained on our dataset and evaluated for its predictive accuracy. We calculated the log loss, which was 0.5713599920272827, indicating the model's efficacy. Additionally, we generated a normalized confusion matrix, visually represented through a heatmap, to better understand the model's classification accuracy between 'Clicked' and 'Not Clicked' categories. This XGBoost implementation marks a significant stride in our endeavor to accurately predict online ad click-through rates. Here is the confusion matrix ***(Figure 8]***

☐ **Testing & conclusion:**

Upon rigorous training and evaluation, our XGBoost model has emerged as the superior predictive model for our project. The results, as illustrated in the provided confusion matrix, underscore its robustness in classifying click-through events. Notably, the model has demonstrated a commendable proficiency in identifying 'Clicked' instances, with an accuracy of ***80.80%*** for true positive rates. This is indicative of the model's effectiveness in discerning patterns that lead to ad clicks.

The confusion matrix also reveals that while 42.22% of the 'Not Clicked' predictions were false positives, the model's ability to correctly predict 'Clicked' outweighs this by a significant margin, making it the most favorable model we have tested. This high true positive rate is crucial for our application, where capturing the instances of ad clicks is more valuable for optimizing ad placement strategies and understanding user engagement.

In conclusion, the XGBoost model's high true positive rate, as highlighted by the heatmap, makes it the best-performing model in our suite of algorithms. It successfully captures a higher proportion of click-through rates, which is paramount in the context of online advertising effectiveness.

Part 5: Recommendation:

The extensive size of our dataset posed a challenge for processing, leading us to use data sampling for manageability. This approach, while practical, might not fully represent the dataset's complexity. Additionally, our limited computing resources restricted the scope of hyperparameter tuning, a key factor in enhancing model performance.

For future improvements, we recommend:

1. **Upgraded Computing Power**: Access to better hardware would allow for full dataset utilization, improving model accuracy.
2. **Comprehensive Hyperparameter Tuning**: With more resources, an in-depth tuning process could be conducted to fine-tune the model for optimal performance.
3. **Advanced Techniques**: Investing in advanced machine learning methods could further refine our predictions.

Addressing these computational challenges is crucial for advancing the model's capabilities and achieving higher predictive precision in online ad click-through rates.

Part 6: Appendix:

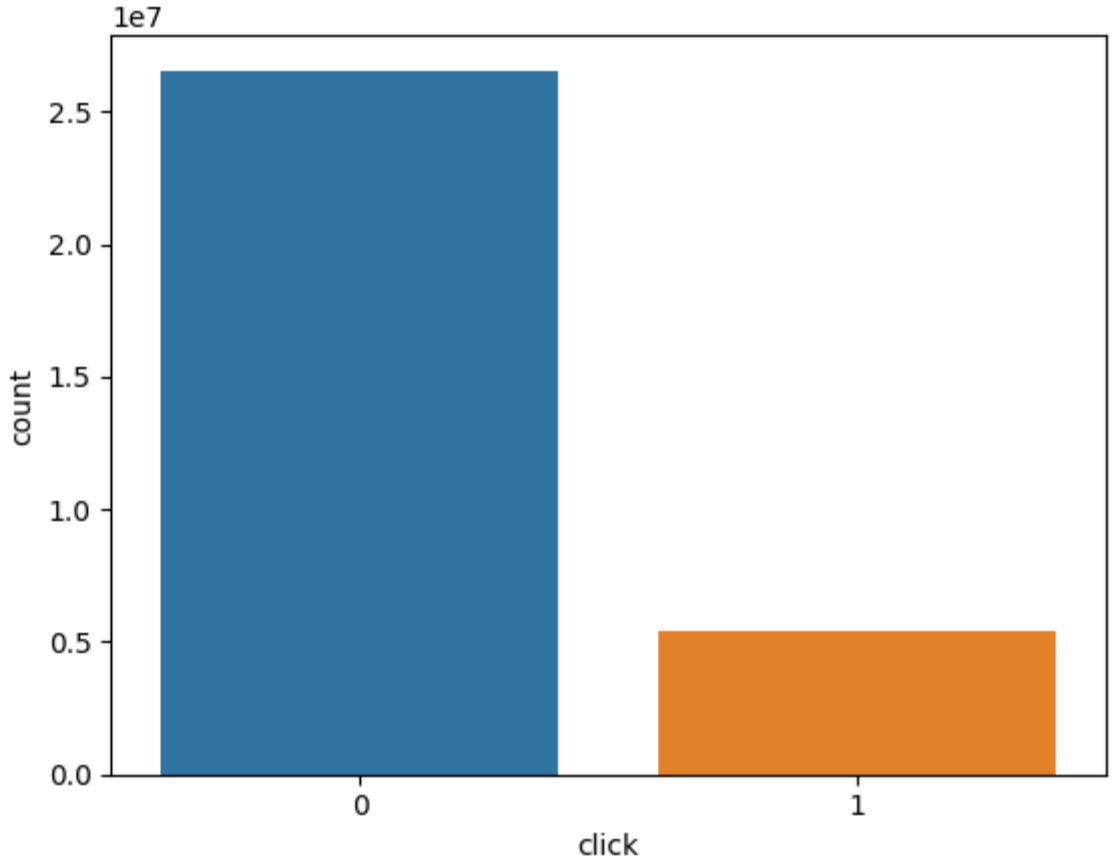

Figure 1: Target Variable

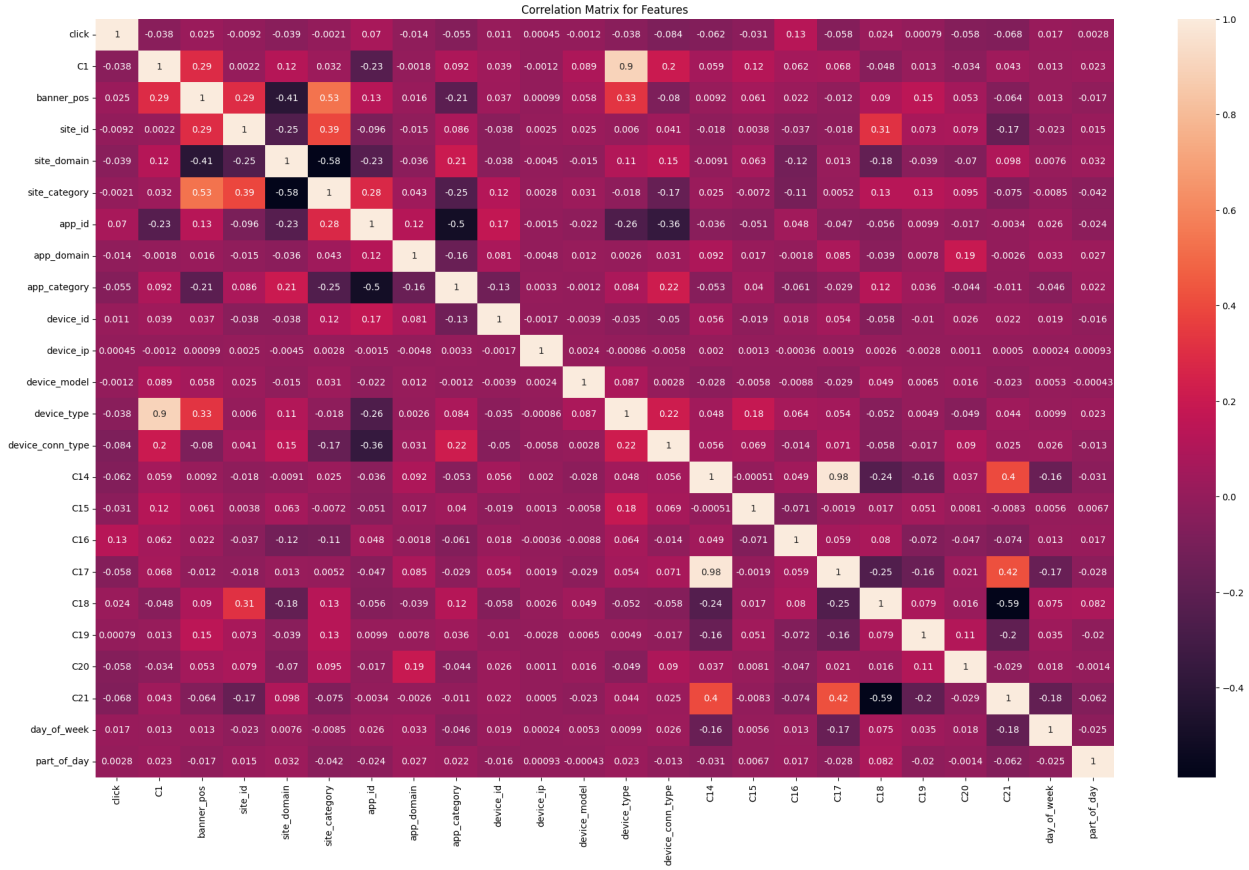

Figure 2: Correlation matrix

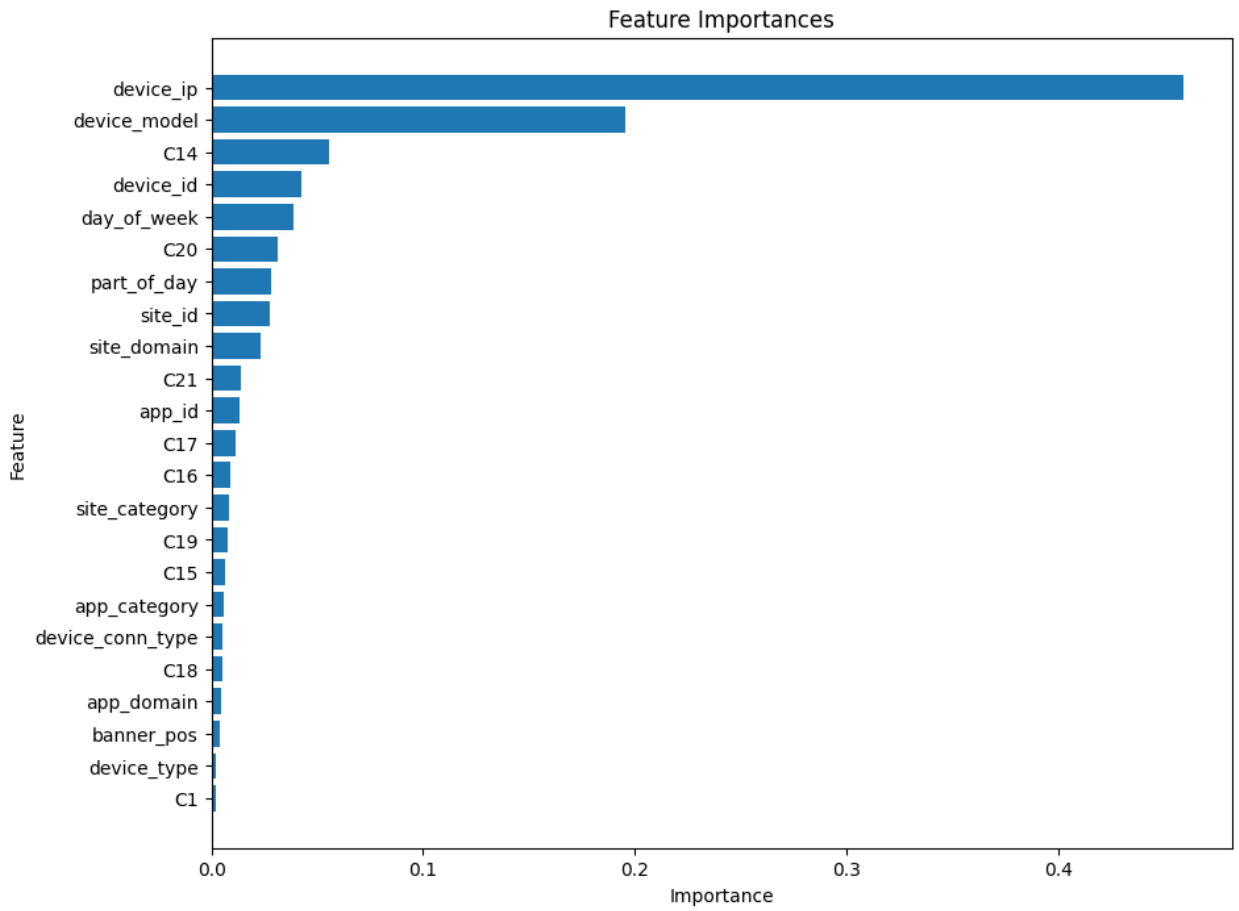

Figure 3: Ranking Based on the feature importance

```
Best parameters: {'n_estimators': 300, 'max_depth': 20}
Best score (log loss): 0.39762833751658744
```

Figure 4: Output from Random Forest after hyperparameter tuning

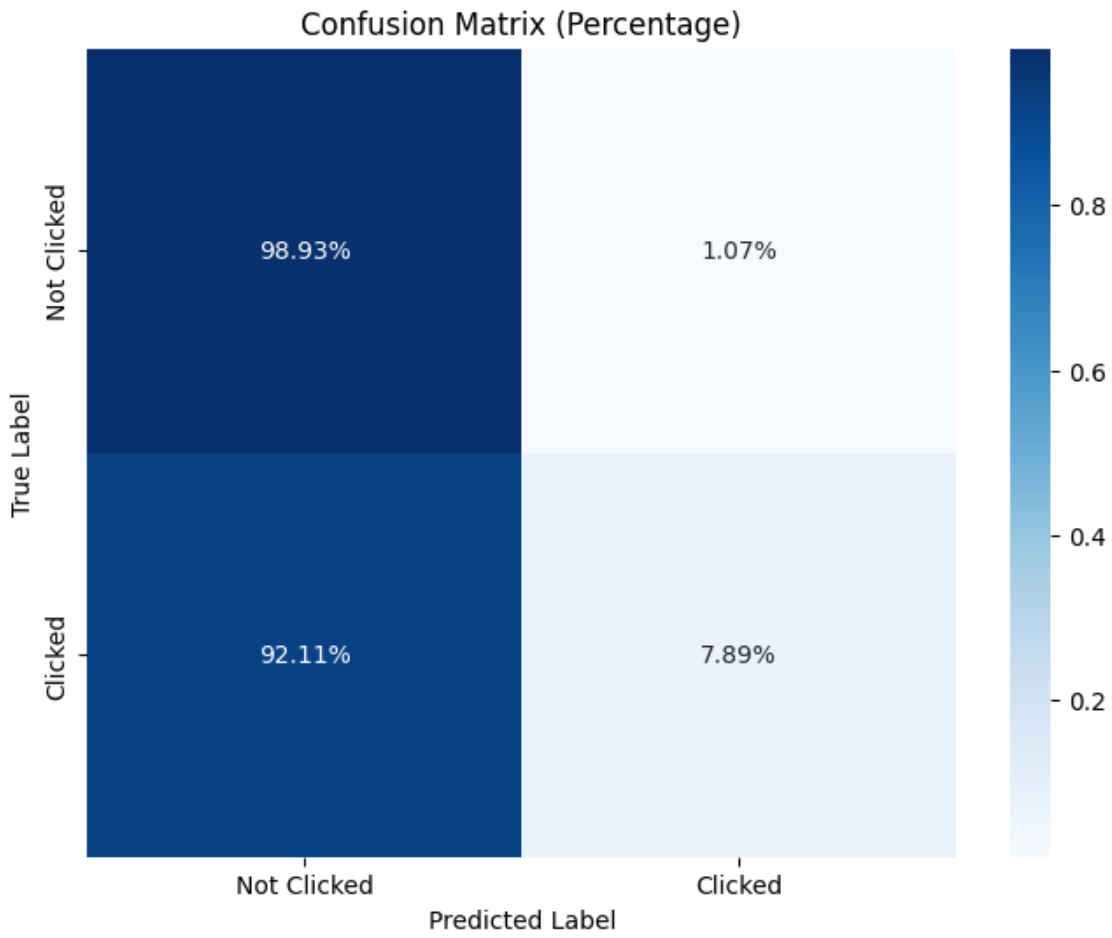

Figure 5: Confusion Matrix

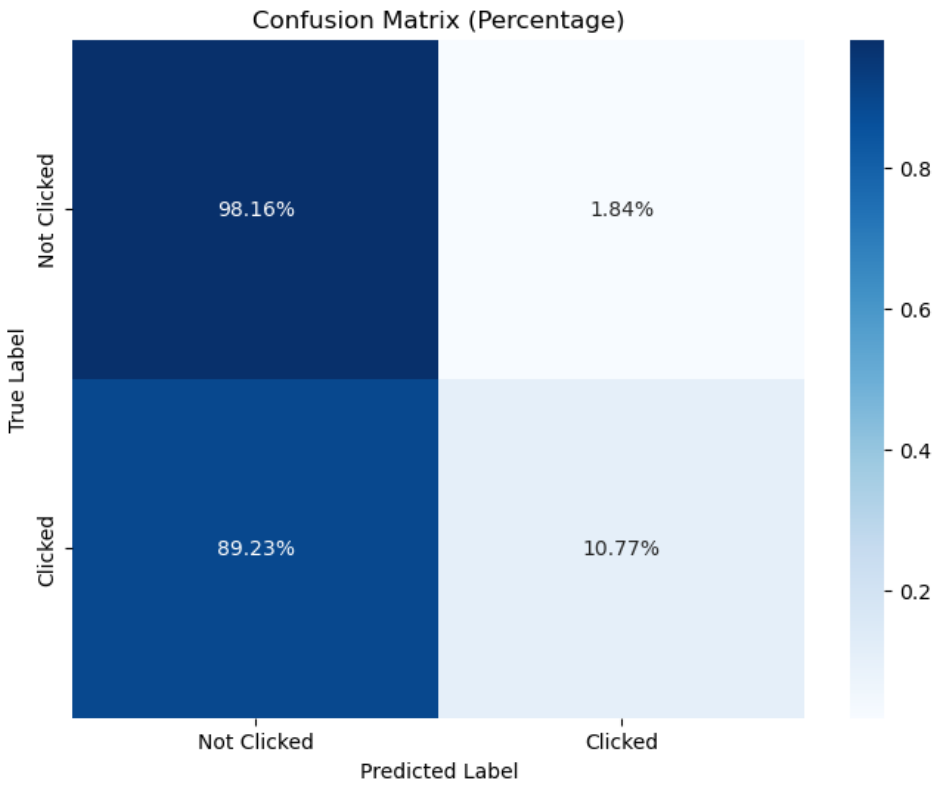

Figure 6: Confusion Matrix

```
=====================>.] - ETA: 0s - loss: 0.6894 - accuracy: 0.5436
not improve from 0.68938
=======================] - 299s 3ms/step - loss: 0.6894 - accuracy: 0.5436 - val_loss: 0.6894 - val_accuracy: 0.5432

=====================>.] - ETA: 0s - loss: 0.6894 - accuracy: 0.5436
not improve from 0.68938
=======================] - 307s 3ms/step - loss: 0.6894 - accuracy: 0.5436 - val_loss: 0.6895 - val_accuracy: 0.5432
```

Figure 7: Neural nets results

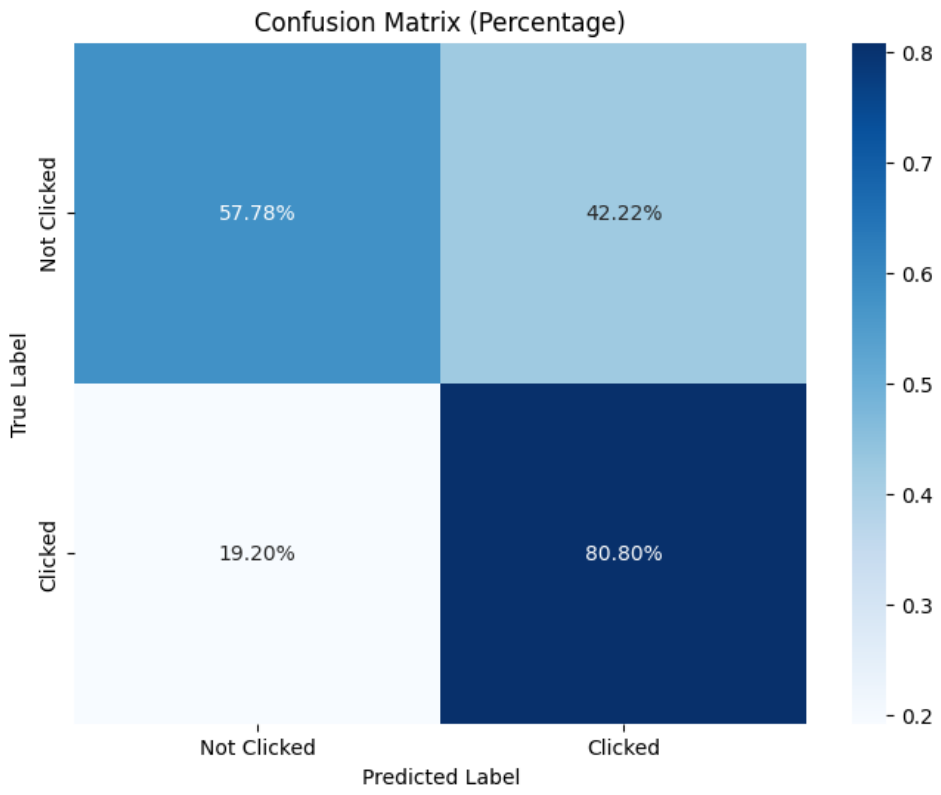

Figure 8: Confusion Matrix for XGBOOST

Code file:

All codes all in one files excerpt XGBOOST MODEL code which is separate in its own folder

File name 1: Final Project.ipynb

File name 2: xg_bost.ipynb