Team members: Ivanie Umuhoza, Zach Wei, Winston Tsai, Jiayu Li

Objective

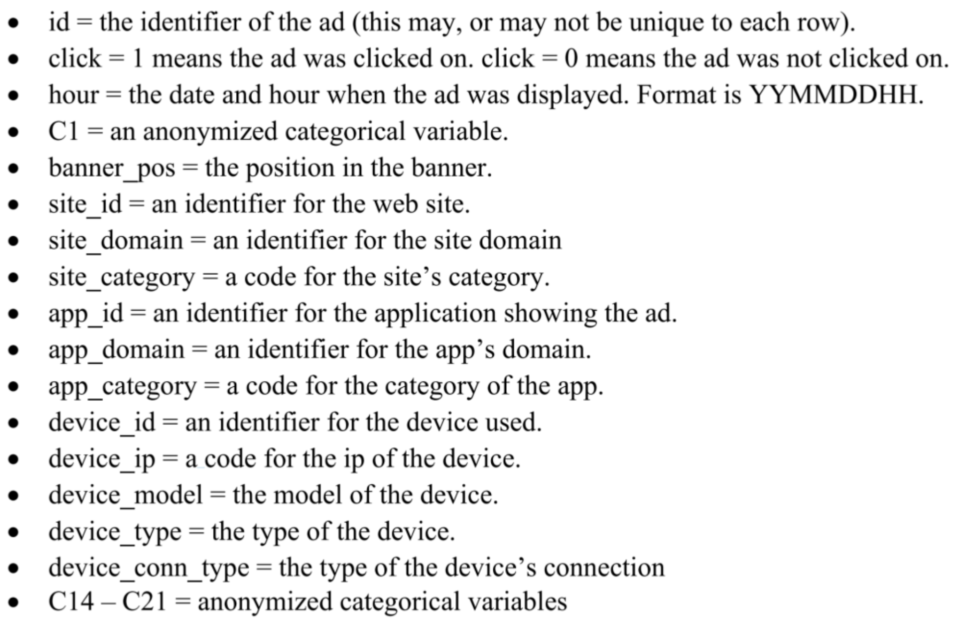
The goal of this project is to build a machine learning model to predict whether an online advertisement will be clicked based on various features of the ad, user device, and website context. The features are explored in the dataset overview, and the model we have is trained to minimize the log loss metric.

Dataset Overview

The training dataset contains over 30 million records, each representing an ad with the following features:

- **Categorical Variables**: Ad identifiers, site domain, app identifiers, device model, and anonymized features (C1, C14 to C21).
- **Numerical Variable**: The hour the ad was displayed, transformed into weekday and hour.
- **Target Variable**: click, where 1 indicates the ad was clicked and 0 indicates it was not.

Due to the dataset's size, a 10% random sample of the training data was used for model development. An overview of all the features in R can be found in Appendix A.



- id = the identifier of the ad (this may, or may not be unique to each row).
- click = 1 means the ad was clicked on. click = 0 means the ad was not clicked on.
- hour = the date and hour when the ad was displayed. Format is YYMMDDHH.
- C1 = an anonymized categorical variable.
- banner_pos = the position in the banner.
- site_id = an identifier for the web site.
- site_domain = an identifier for the site domain
- site_category = a code for the site's category.
- app_id = an identifier for the application showing the ad.
- app_domain = an identifier for the app's domain.
- app_category = a code for the category of the app.
- device_id = an identifier for the device used.
- device_ip = a code for the ip of the device.
- device_model = the model of the device.
- device_type = the type of the device.
- device_conn_type = the type of the device's connection
- C14 – C21 = anonymized categorical variables

Data Preprocessing

- **Feature Selection and Removal**: The id, device_ip, and device_id columns were removed as they do not contribute meaningfully to the prediction task. The column variables that were kept were more valuable for the prediction task and had stronger contributions to the final predictions. For example, the variable banner_pos shows directly how the position of the ad would impact the final probability of a click. The scores for the feature selection found by running R code can be found in Appendix B.
- **Feature Transformation**: The hour column that was originally in the YYMMDDHH format, was transformed into two new numeric features:
    - **weekday**: The day of the week (1 for Monday to 7 for Sunday).
    - **hour**: The hour of the day (0 to 23).
- **Categorical Encoding**:
    - The categorical variables, including anonymized features C14 to C21, were grouped into an "others" category for low-frequency values to reduce sparsity.
    - All categorical variables were also converted to factors using as.factor for compatibility with the CatBoost library.

Feature Importance

The feature importance analysis for predicting clicks on online advertisements reveals key insights about the model's decision-making process. Among the categorical variables, features such as C18 and C21 hold the highest importance scores, suggesting they play a crucial role in distinguishing between clicked and non-clicked ads. Other significant contributors include C19, C16, and site_id, indicating that the anonymized categorical variables and the specific identifiers for sites strongly influence the probability of a click. The importance of app_category and banner_pos underscores the relevance of contextual placement and application type in user engagement. Conversely, variables like weekday show no importance, suggesting they have minimal or no predictive power in this dataset. Overall, the results guide optimization efforts, emphasizing the need to focus on the most influential features to improve ad targeting strategies.

Model Selection

The CatBoost library was chosen for its native handling of categorical features and efficiency with large datasets since it does not require much more processing. It also helps prevent overfitting and the problem with high memory consumption that destroys our computers.

The model was configured to minimize log loss as the evaluation metric, which for our project is more well suited for this problem since it deals with a supervised classification problem. Minimizing the log loss in the CatBoost model aims to maximize the accuracy and reliable predictions, which helps people who make the analysis decisions make the right choices.

Hyperparameter Tuning

Grid search was performed to optimize the following hyperparameters:

- **Depth**: Maximum tree depth (4, 6, 8).
- **Learning Rate**: Step size for optimization (0.01, 0.1, 0.2).

For each combination of hyperparameters, the model was trained on an 80% training split and evaluated on the remaining 20% validation split. The log loss was calculated on the validation set for each configuration. The best hyperparameters for iterations, depth, and learning rate are:

- **Iterations**: 1000
- **Depth**: 8
- **Learning Rate**: 0.2

Final Model Training

Using the best hyperparameters, the final CatBoost model was trained on the full sampled training data and validated on the 20% validation set. The final model achieved the lowest log loss on the validation set. The model was based off of first of all, the 10% sample taken out of the large dataset because of the size, and then split into 80% test and 20% validation. After the model is trained, we can apply it to the test data in order to see the results of the model

Model Importance

The predictive model we created is crucial in a business context because it provides actionable insights into ad performance, enabling companies to make data-driven decisions. By accurately predicting the likelihood of an ad being clicked, the model allows businesses to optimize their advertising spend, target the right audiences, and improve campaign efficiency. This leads to better ROI, higher user engagement, and more effective resource allocation. Additionally, the model's ability to handle large datasets and minimize log loss ensures that predictions are both scalable and reliable, making it a valuable tool for driving strategic business outcomes in the competitive world of online advertising.

Testing

The model was applied to the test dataset csv that was given, and the click probabilities were predicted in a separate column from the test data using the model we trained. If ground truth labels were available for the test dataset, the log loss metric could be calculated.

Results

- **Validation Log Loss**: The lowest log loss achieved during hyperparameter tuning.
- **Test Predictions**: Probabilities for each ad being clicked were saved to a file named TestPredictions.csv.

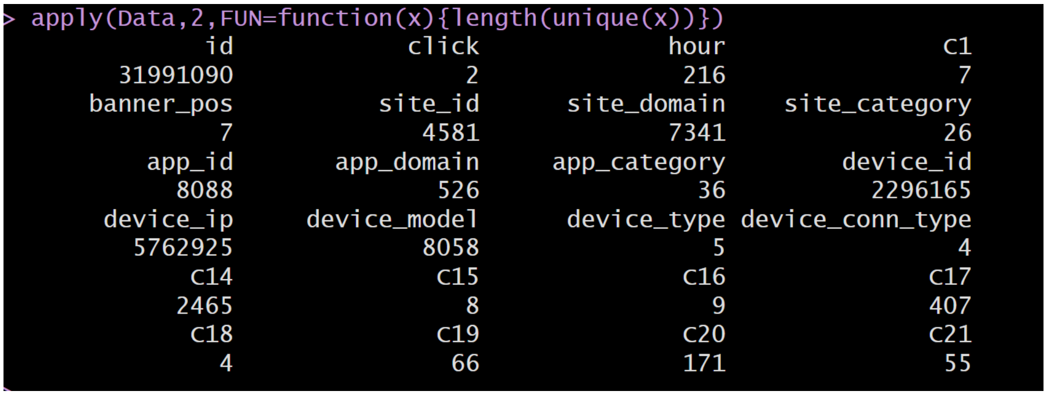
Future Work

- **Feature Engineering**: Additional transformations or feature creation could improve model performance.
- **Model Ensemble**: Combining CatBoost with other models such as XGBoost or LightGBM could yield better results.
- **Deployment**: The model can be deployed in a real-time system to predict ad clicks and optimize advertising strategies.
- **Bootstrap:** Utilize bootstrap samples instead of just a 10% sample to first of all capture more of the data and capture the overall distribution of the data better.

Conclusion

This report outlines a systematic approach to building a predictive model for ad clicks. Through careful preprocessing, hyperparameter tuning, and validation, a robust CatBoost model was developed to minimize log loss, making it suitable for real-world application in online advertising. This model can be applied to more data that is gathered and the results can be analyzed for further use.

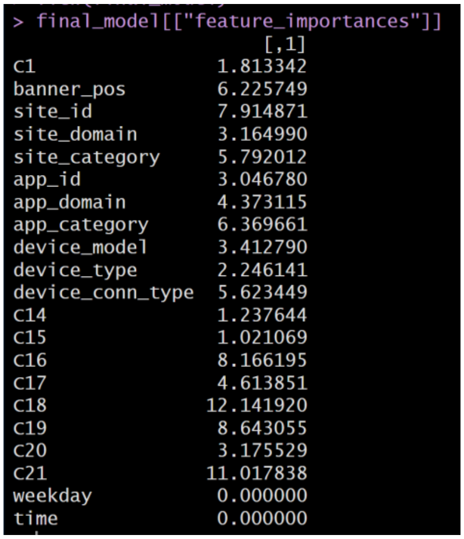
Appendix A



```
> apply(Data,2,FUN=function(x){length(unique(x))})
              id            click             hour               C1
        31991090                2              216                7
      banner_pos          site_id      site_domain    site_category
               7             4581             7341               26
          app_id       app_domain     app_category        device_id
            8088              526               36          2296165
       device_ip     device_model      device_type device_conn_type
         5762925             8058                5                4
             C14              C15              C16              C17
            2465                8                9              407
             C18              C19              C20              C21
               4               66              171               55
```

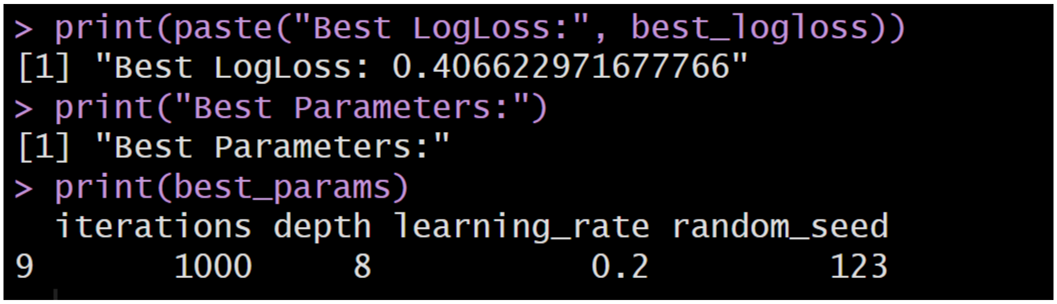
Appendix B

Code for feature selection, higher scores mean more contribution to the final prediction model

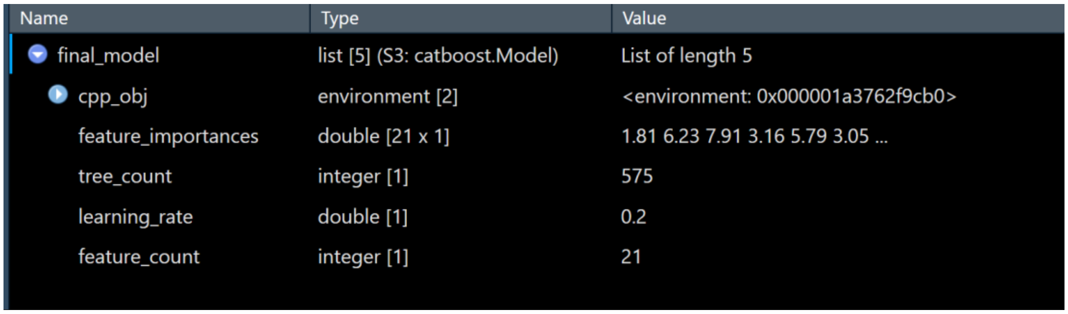


```
> final_model[["feature_importances"]]
                       [,1]
C1                 1.813342
banner_pos         6.225749
site_id            7.914871
site_domain        3.164990
site_category      5.792012
app_id             3.046780
app_domain         4.373115
app_category       6.369661
device_model       3.412790
device_type        2.246141
device_conn_type   5.623449
C14                1.237644
C15                1.021069
C16                8.166195
C17                4.613851
C18               12.141920
C19                8.643055
C20                3.175529
C21               11.017838
weekday            0.000000
time               0.000000
```

Hyperparameter selection using grid search, the results from the code are below.



```
> print(paste("Best LogLoss:", best_logloss))
[1] "Best LogLoss: 0.406622971677766"
> print("Best Parameters:")
[1] "Best Parameters:"
> print(best_params)
  iterations depth learning_rate random_seed
9       1000     8           0.2         123
```



Name	Type	Value
final_model	list [5] (S3: catboost.Model)	List of length 5
cpp_obj	environment [2]	<environment: 0x000001a3762f9cb0>
feature_importances	double [21 x 1]	1.81 6.23 7.91 3.16 5.79 3.05 ...
tree_count	integer [1]	575
learning_rate	double [1]	0.2
feature_count	integer [1]	21