MSBA Team 5: Henry Dimlow, Pandora Shou, Zayn Sui, Cindy Zhang

ISOM 674 - Machine Learning Project Write-up

Introduction

For our machine learning project, we aim to predict the possibilities of an online-advertisement being clicked. We deployed logistic regression, XGBoost, random forest classifier, LightGBM, and the CatBoost model. After performing hyperparameter tuning on the training data and comparing the log loss of each model, we eventually decided to choose LightGBM as our final best model, with a log loss of only .4084.

Data Preparation

Before starting work on our model, we observed the data and found that there was a need to manipulate our data to work with it more easily. We found several columns that needed some sort of data manipulation:

- First, we deconstructed the hour column into 4 columns with year, month, day, and hour so that we could examine each of these variables more closely
- We dropped the year and month columns as all data had these values in common
- We then created a weekday column by encoding each weekday into a value. The weekday values increase with each day where Monday is 0 and Sunday is 6
- For each column we only included values if they appear in the data more than 1% of the time in order to reduce the dimensionality of the dataset and keep the models from trying to overfit based on features that are likely mostly irrelevant.

- We removed id, device_ip, and device_id from the dataset as these have little or no predictive power and are used as identifiers for the data.

Models and Evaluation

In terms of modeling, we initially decided to sample 10% of the total data from ProjectTrainingData.csv. However, none of our computers can handle 3 million rows of data in parameter tuning and it significantly influenced the whole project process. Therefore, we decided to sample 1% of the whole data to conduct parameter testing and evaluate the models.
After sampling 1% of the total training data, we created a new training data containing 80% of the sampling data, and a new validation data containing 20% of the sampling data. We evaluated the model by fitting the validation data.

- Logistic Regression
    - We performed parameter tuning and sample prediction on the training sample
    - We tuned parameters C and penalty resulting in:
        - C = 1
        - Penalty =‘ L1’
    - We then fit the model using the training data
    - Predicted probabilities on the test set using our model and checked log loss resulting in a log loss of .4364.

- XGBoost
    - We performed parameter tuning and sample prediction on the training sample
    - We tuned minimum child weight, maximum depth, and learning rate resulting in:

        - learning_rate = 0.25
        - max_depth = 6
        - min_child_weight = 1
    - We then fit the model using the training data
    - Predicted probabilities on the test set using our model and checked log loss resulting in a log loss of .4290.

- Random Forest Classifier
    - We performed parameter tuning and sample prediction on the training sample
    - We tuned minimum samples per leaf, number of estimators, and the max depth resulting in:
        - Min_samples_leaf = 20
        - N_estimators = 1800
        - Max_depth = 60
    - We then fit the model using the training data
    - Predicted probabilities on the test set using our model and checked log loss resulting in a log loss of .4085 for our sample

- LightGBM
    - We performed parameter tuning and sample prediction on the training sample
    - We tuned parameters num_leaves , min_data_in_leaf , and reg_alpha, bagging_fraction, and feature_fractions resulting in:
        - Num_leaves = 70

        - Min_data_in_leaf = 50
        - reg_alpha = 2
        - Bagging_fraction = 0.1
        - Feature_fraction = 0.5
    - We then fit the model using the training data
    - Predicted probabilities on the test set using our model and checked log loss resulting in a log loss of .4084 for our sample

- Catboost
    - We performed parameter tuning and sample prediction on the training sample
    - We tuned learning rate, depth , and iteration resulting in:
        - learning_rate = 0.28
        - N_estimators = 8
        - Max_depth = 62
    - We then fit the model using the training data
    - Predicted probabilities on the test set using our model and checked log loss resulting in a log loss of .4109 for our sample

Final Model: LightGBM

After testing and comparing our models, we decided to use LightGBM because the log loss of .4084 was better than any other model, slightly beating the random forest classifier.

Appendix

Code Files:

FinalProj-ReadProjectData.r

- We used this file to process the train data and test data and for training data to create smaller subsets of the data that could be run more quickly.

modeling_random_forest.ipynb

- We built random forest model and tuned values of parameters and evaluated performance using log loss.

modeling_logistic_regression.ipynb

- We built logistic regression model and tuned values of parameters and evaluated performance using log loss.

modeling_LightGBM-XGBoost-Catboost.ipynb

- We built LightGBM, XGBoost and Catboost models and tuned values of parameters and evaluated performance using log loss.