Summary

In this project we analyzed the clickstream data of an ecommerce retailer and built robust classifiers in order to predict the session to be a buying session or a non-buying session and further predicted the items customer would buy if the session would be a buying session. We did sufficient feature engineering and resampling of the unbalanced data before feeding our data to the models. In this project we reviewed traditional and advanced machine learning approaches to address this binary classification problem. We started with the base model of Logistic regression, later to improve the accuracy of the classifier we implemented ensemble learning algorithms with Random Forest and Light Gradient Boosting. With this ensemble learning approach, we improved over logistic regression and also over deep learning algorithm. Light GBM outperformed and was the best classifiers with maximum AUC score of 0.79. Further we addressed the second objective the project of identifying the potential items customer would buy based on his clicks using the KNN approach by identifying the similarity in the clicked items.

Thus, we observed that by unifying the clickstream data and understanding the customer behavior metrics it is possible to predict entire journey of the customer on the platform.

Introduction

Clickstream data are the detailed logs or a record of the user actions while traversing through a particular website. This data has been a great source for analyzing the customer behavior and is widely studied across multiple industries. Once such industry where the clickstream data plays a vital role and which is widely validating the customer's clickstream data to understand the customer behavior is the ecommerce industry. Ecommerce clickstream data can be helpful in identifying customer insights, identifying the popular products and product specific data. Using this data ecommerce industries are delivering the top n recommendations to the customers, which is the ultimate strategy to up scale the customer experience and in a way decide on strategies to boost the buying events. This encouraged us in choosing the clickstream data for the ecommerce domain as our topic for the course final project.

As part of this project we analyzed clickstream data for a large-scale ecommerce retailer in Europe which was published as part of the RecSys 2015 challenge. We implemented a solution to identify the user intent for a particular session and predicting whether a customer would be making buying an item or not. We further predicted what items user would be potentially purchasing if a particular user session intends for a buying event.

This analysis can be further be used by the marketing team in building robust strategies to boost the product sales and planning over improving the customer experience in order to optimize the conversion rates.

Goal

With a sequence of clickstream events performed by a user on an e commerce website, following are the business outcomes which would be addressed.

1.Predicting whether the customer would be making some purchase or not.

2.If yes what are the items that are going to be bought.

Data Overview

The dataset for the analysis includes 2 files, yoochoose-clicks.dat and yoochoose-buys.dat.

The yoochoose-clicks.dat reports :

Session ID	the id of the session
Item ID	the unique identifier of the item
Category	the category of the item
Timestamp	the time when the click occurred

The file yoochoose-buys.dat records the buy events and includes

Session ID	the id of the session
Item ID	the unique identifier of the item
Category	the category of the item
Timestamp	the time when the buy occurred
Price	the price of the item
Quantity	Number of items bought.

	sessions	**clicks**	**clicked items**
Training Set	9249729	33003924	52739

Merging of the click events with the buying events for a particular session together describes the overall activity performed by the customer in a particular session.

Implementation

1. Feature Engineering :

With limited features in the given data, we had to engineer more features from the data before starting with our analysis. Following were the features which we have engineering before starting with the predictions.

Features by Session:

-Session Start time
-Session end time
-Total Session Time
-Total Session Time
-Total No. of Clicks in a Session
-Day of Click
-Month of Clicks
-Hour of Click
-Time of Day
-Click Rate
-Buy/Not Buy(**Target**)

Features by clicked Items & Categories

-First Clicked Item
-Last Clicked Item
-Total Unique Items
-Total Unique Categories
-Visited Items
-Visited Categories
-Count of Visited Categories
-Special offer clicked
-Total number of items bought
-Unique items which were bought
-Popularity of the first clicked item
-Popularity of the last clicked item
-Popularity Click Index for each item
-Popularity Buy Index for each item

While exploring the data we could find certain trends in the data revealing some important customer behaviors in the data.

1.Maximum clicks where observed on special offers.

2.A total of 509,696 positive sessions are available in the training set reporting a buying event.

3. Average dwell time for a positive buying event is 867.10 sec ~ 14.45 mins.

4. Average dwell time for a negative buying event is 353.32 sec ~ 5.8 mins.

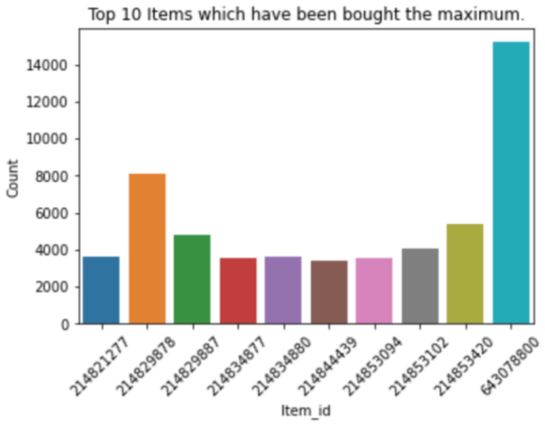
5.Maximum buying events have been observed on Sundays and Mondays.

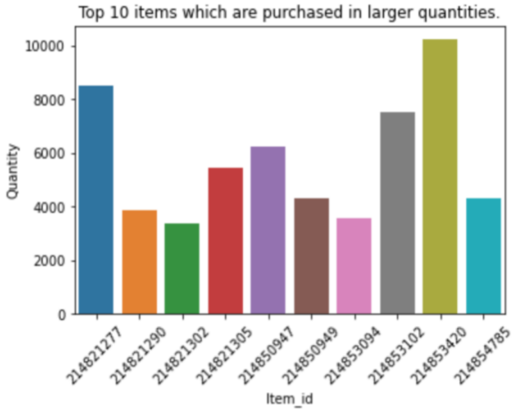
6.Worst buying events are observed on Tuesdays.

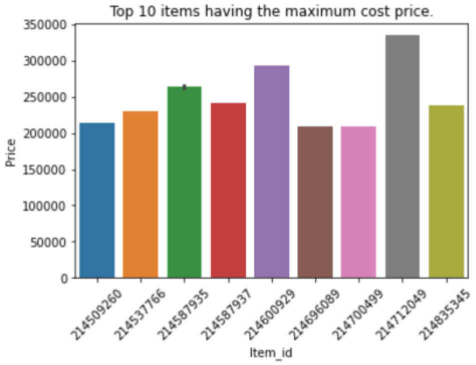
7.Evening time is suitable for a greater number of buying events.

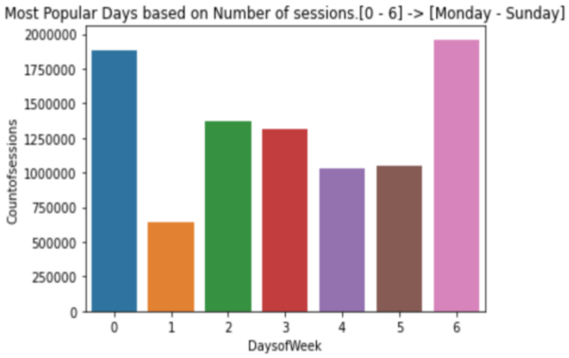
8.Also we found that the of all the buying events,70% of times the item which was clicked first and 72% of times the item which was last clicked were purchased by the customer.

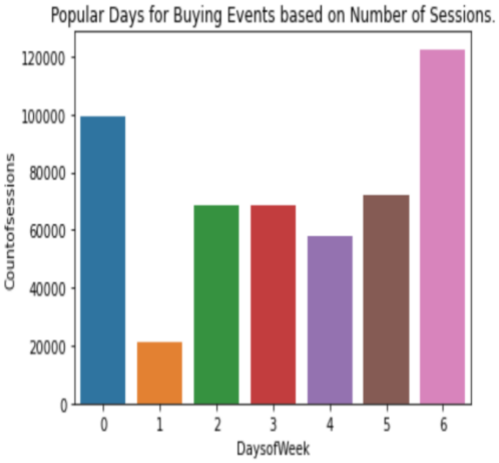
Analysis of Buying Events:

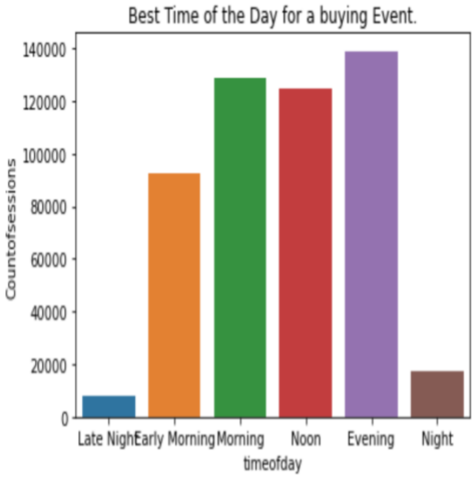


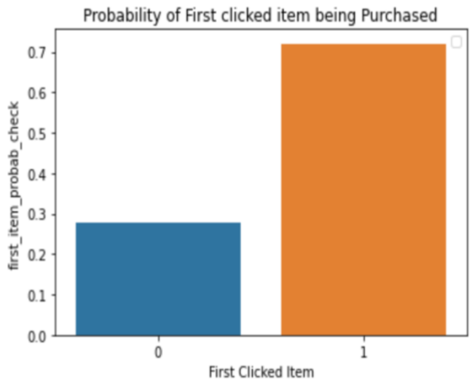


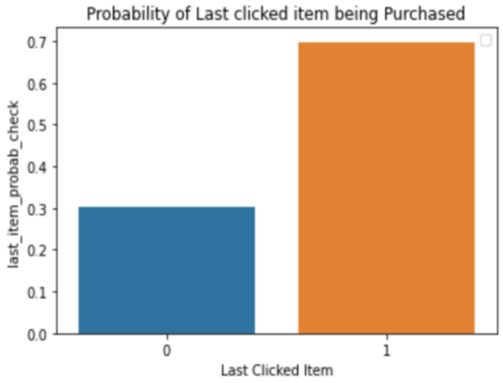












2. **Handling Unbalanced data:**

The buys data provided reports a total of 509696 positives buying events which constitute towards only 5% of the of the total available dataset. Thus, one of most important obstacles before proceeding to modeling of the data was handling of the unbalanced classes. In addressed this by random under sampling of the data. Random under sampling is the technique in which the class with the majority class is reduced to match the minority class.

Thus, with this we balanced the ratio of classes in the data to a balanced state.

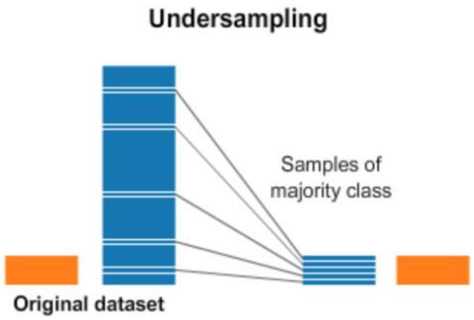


Figure 1:Random Under Sampling.

3. **Splitting of Data:**

Before we start the data modelling, we split the data into training and testing datasets in 80:20 ratio in order to efficiently validate the model’s performance.

Data Analysis :

Our problem statement is divided into two tasks as follows:

Task 1: Predicting the Session to be a buying event or non-buying event.

Analysis for Task 1:

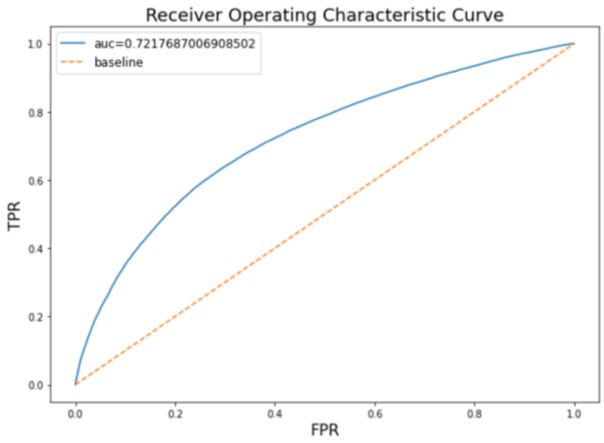
This is a type of supervised learning task where the categories are predefined with binary labels thus a binary classification machine learning problem.

1.Logistic Regression:

Logistic regression is a statistical model which uses the logistic function to model the binary dependent variable. Since the learning tasks of predicting the buying event for a session is a binary classification task, we have chosen logistic regression as our first base model to start with data modeling. Logistic regression cannot handle non integer data and our features into consideration included some of the categorical features which were not integer encoded thus there was a need to one hot encoded these categorical features before feeding the features to the logistic regression model.

1.2 Modeling Results:

Below are the model execution results for predicting the whether a customer would be making a buying event in a session. Our Logistic model predicted with AUC score of 0.72 on the test data with a sensitivity of 0.64 and specificity of 0.63 for an optimal threshold value of 0.44. Below is the ROC curve metric describing the AUC and the tradeoff between the FPR and TPR at different thresholds values.



Sensitivity	Specificity
0.64	0.63

	Predicted 0	Predicted 1
Actual 0	64889	36896
Actual 1	36333	65763

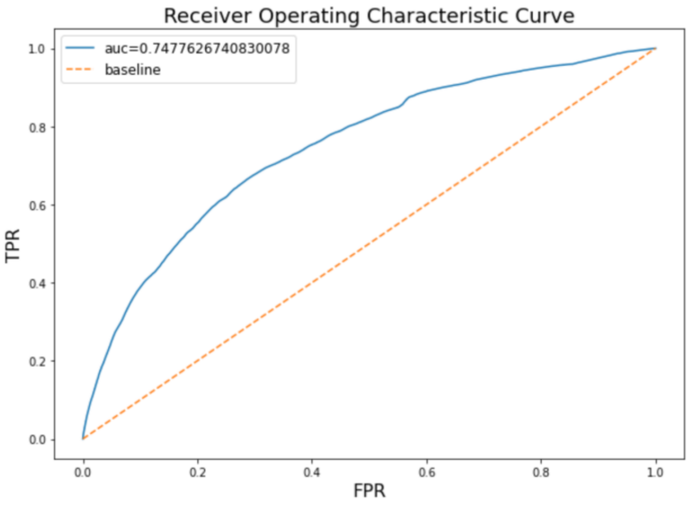
The accuracy of the logistic regression model is not convincing and model resulted in false results, thus we implemented Random Forest Classifier to improve the performance of the classifier.

2.Random Forest :

Random Forest Classifier is a classification algorithm consisting of multiple decision trees and uses a bagging technique where the final outcome is a combination of the multiple learning models of decision trees. The output of the random forest is a mode of the results given by the multiple decision trees. The multiple decision trees which are created here are trained independently using the random subset of the training dataset.

2.1 Modeling Results:

Below are the execution results obtained for the Random Forest Model execution. A considerable improvement in the classifier results have been observed with random forest over logistic regression. AUC score of 74.77 , 2% rise in the performance over logistic regression. These results were obtained with hyperparameter tuning for the number of the decision trees = 200 and a maximum depth = 4.

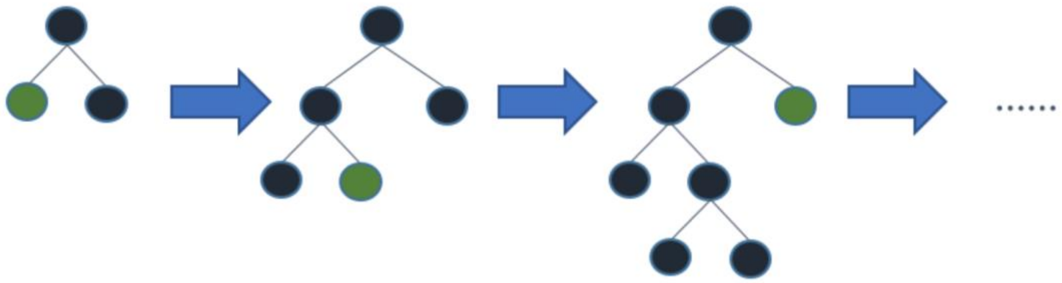


	Predicted 0	**Predicted 1**
Actual 0	73012	28773
Actual 1	34766	67330

Sensitivity	**Specificity**
0.71	0.65

3. Light Gradient Boosting Machine:

It is a fast gradient boosting algorithm which is based on decision tree algorithm and another type of ensemble learning algorithm. The idea behind the Gradient Boosting Decision trees is combining the outcomes of the multiple decision trees where each tree is training multiple iteratively to minimize a loss function. While other boosting algorithms split the trees based on level the LGBM splits the tree leaf wise which makes the algorithm execution to be fast.



Leaf-wise tree growth

2.1 Advantages of using LGBM over this data

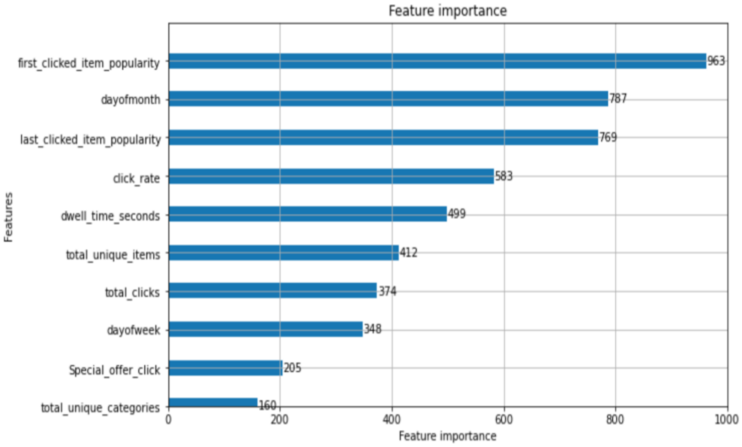
1. Lower memory usage:
2. Compatibility with Large Datasets

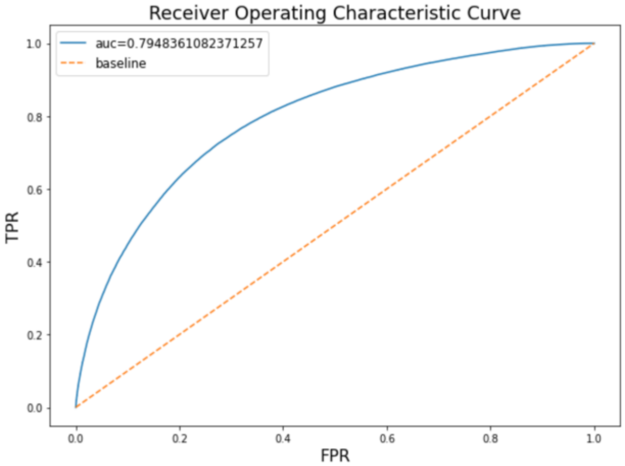
2.2 Modeling Results:

Below are the model execution results for predicting the whether a customer would be making a buying event in a session. Our LGBM model predicted with AUC score of 0.79 on the test data thus performing better than Random Forest and giving out balanced results with sensitivity and specificity. These results were obtained with hyperparameter tuning of number of trees = 150, num of leaves = 35 and a learning rate = 0.35 using cross validation techniques. Below plots describe the important features identified by the model and the ROC curve for the model execution.

	Predicted 0	Predicted 1
Actual 0	73608	28177
Actual 1	27911	74185

Sensitivity	Specificity
0.723	0.72





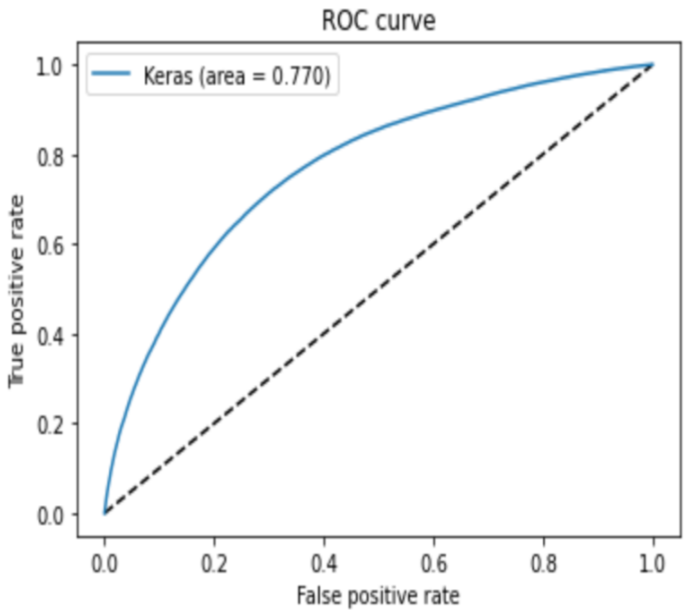
We further implemented deep learning over the data to validate in order to further improve the accuracy of the classifier.

4.Neural Network:

Neural Nets is a means of doing machine learning in which the algorithm learns to cluster and classify the data based on the similarities of the input data. Neural networks are composed of multiple layers that are made up of nodes. A node is where the computation is done which combines input from the data with a set of weights that amplifies or dampens the computation output. The input weight products from multiple nodes is combined and passed through an activation function to understand to what extent the output should be processed further.

Optimal Neural Network:

We implemented a Neural Net with a ReLu activation function of 18 inputs and a binary output with sigmoid activation function at the output layer. We hyper tuned the model for multiple validations of the parameters but the best hyper tuned model obtained was a learning rate of 0.0006 with 2 hidden layers of 6 and 5 nodes each with ReLu activation. With this model we could achieve an AUC score of 77.6.



Sensitivity	**Specificity**
0.73	0.69

Task 2: If session is predicted to be a buying event predict the potential items customer would buy based on his clicks.

Analysis of Task 2:

As there were no user id's in the data to build a connection in the session and therefore applying collaborative filtering models was not feasible to identify the potential items customer would buy. Thus, we focused on the available features which we engineered before in our solution.

K Nearest Neighbors:

As a solution for the task 2 we implemented K Nearest Neighbors and calculated the distance between the multiple events in a session and found the most similar 2 events in session using the Minkowski distance with a p value of 2. Smaller the score, greater the similarity. We used the following features for calculating the distance between the events.

- Session
- Popularity Click Index for each item
- Popularity Buy Index for each item
- visited item
- visited category
- special offer

Validation of the Items Predictions:

We performed our validation on the 5000 samples from the training data as it was computationally expensive to iterate over the whole training data and calculating the distances for each session. We observed a prediction accuracy of 77% where we checked if 2 predicted items for each session existed in a purchased item list for a session in the training data.

Conclusion:

In this final project report, we described our analysis for the clickstream data of an ecommerce retailer which was provided as part of the RecSys 2015 challenge. We implemented 4 classifiers starting with the traditional base model of logistic regression and further implementing more robust models with ensemble learning techniques of bagging and boosting with Random Forest Classifier and Light Gradient Boosting respectively. We observed a significant improvement in the model performance using Light Gradient Boosting algorithm. With LGBM we could achieve an AUC score of 79.48 and a sensitivity - specificity rate of 0.72.We further implemented deep learning over the data by fine tuning the parameters of the neural network but we could achieve a maximum AUC score of 0.77. Thus, of the 4 classifiers Light GBM outperformed and was the best classifier with better predictions.

	AUC	SENSITIVITY	SPECIFICITY
LOGISTIC REGRESSION	0.72	0.64	0.63
RANDOM FOREST CLASSIFIER	0.74	0.71	0.65
LIGHT GRADIENT BOOSTING	0.79	0.723	0.72
NEURAL NETWORK	0.77	0.73	0.69

Further we built a model for the task 2 of the problem objective using the K Nearest Neighbors approach to predict the items customer would potentially buy if the session is predicted to be a buying session. We validated over 5000 samples of the sessions from the training data and we observed an accuracy of 77%.

Discussion:

With this analysis of ecommerce clickstream data, we could find how predictable human behavior is and how a machine learning model can identify similarity in the input features to make significant learning from the input data to predict the entire journey of the customer. Further it was surprising for us to see how Gradient Boosting algorithm performed exceptionally well than that of a deep learning model.

Clickstream analytics solution are deployed across multiple industries like that of media and entertainment, banking, travel , hospitality, logistics etc where this data is used to predict the demand for the resources. Advertisement industry is one of the industries where this kind of clickstream data is widely studied to see the human behavior and predicting the potential advertisements a particular user would click on.

With this project we implemented the course work we did in the last 6 weeks, on a real-world problem thus allowing us improve our understanding towards different machine learning algorithms and complementing our course work.

References:

1. RecSys Challenge 2015.Retrieved from https://recsys.yoochoose.net/challenge.html

2. (December 18,2018) Koehna D., Lessmanna S & Schaalb M. Predicting online shopping behaviour from clickstream data using deep learning. https://doi.org/10.1016/j.eswa.2020.113342

3. (2015). Ben-Shimon, Tsikinov sky, Friedmann, Shapira, Rokach & Hoerle .RecSys Challenge 2015 and the YOOCHOOSE Dataset. Association of computing Machinery.10.1145/2792838.2798723.

4. (June 12 2017).P Khandelwal. Which algorithm takes the crown. Light GBM vs XGBoost. AnalyticsVidhya. Retrieved from https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/