Customer Revenue Prediction

STAT 5600: Methods of Statistical Learning

Sujan Barama
sujan.barama@colorado.edu

Thejas Kiran
thejas.kiran@colorado.edu

Customer revenue prediction is one of the important fields that most companies in all industries are looking to expand. The companies would want to earn higher revenue from a set of highly loyal customers as the revenue for a production company follows the Pareto principle i.e, 80% of the revenue for a company comes from 20% of its customers. We follow a different approach to generate these forecasts of revenues. We first generate a classification forecast telling whether the person has made a buy or not and then generate the actual value of the transaction using regression algorithms. We will be working on the Google Merchandise e-commerce store to implement this approach. We use the RMSE score as an evaluation metric for our models, but we plan on building a custom accuracy metric as a future scope.

Keywords: *Customer Revenue Prediction, Linear Regression, Logistic Regression, XGBoost Models, Decision Tree Models, and Customer Lifetime Value*

Introduction

The relationship between the customer and the company is particularly important for any company to sustain in their respective industries as the cost of acquiring new customers is more than retaining existing loyal customers. Companies need to figure out the optimal marketing techniques to earn more revenue from customers and minimize the cost of acquisition. This problem is prevalent in most of the companies in all sectors, and they are trying to achieve this using different Machine Learning and Deep Learning techniques. These predictions will help the companies,

- Identify high-value and loyal customers
- Optimize marketing strategy to allocate budget more accurately for customer outreach
- Guide product development to what the user requires
- Create customer segments to for personalized campaigns with target sales

The main goal of our project is to build a Machine Learning technique that can be helpful in predicting the amount the user might spend when he logs into the e-commerce website. In this project, we will only be focusing on the Google merchandise e-commerce store. We proceed with a new approach where a classification model is first built determining whether there has been a purchase and then generating the forecasts for the revenue from a regression model based on these results.

Related Work

There has been a lot of research in the field of predicting sales on e-commerce sites and we have referred to some of this latest research for our project.

The paper *"Optimizing Sales Forecasting in e-commerce with ARIMA and LSTM Models"* [1] by *Konstantinos N Vavliakis* et. al. explains the difficulties that e-commerce sites face due to sudden increases or decreases in demand for the products. This affects the whole inventory and supply chain network. They have used a very innovative approach for this prediction. They first train the model on the preprocessed dataset and then train the residuals on an LSTM model. This worked surprisingly well as they get better results with this approach than the existing ones.

The dataset that we are using is imbalanced and we referred the paper *"A prediction model for an imbalanced dataset ng machine learning"* [2] by *Owk Mrudala* et. al. which explains the solution to this problem. Although they have worked with medical datasets, we can implement the idea of sampling and decreasing the threshold for classification for our dataset. They have used some basic machine learning algorithms like Random Forest Classification, Logistic Regression, and Decision Trees.

The yet to be published paper *"How using machine learning classification as a variable in regression leads to attenuation bias and what to do about it"* [3] by *Han Zhang* explains the problems of our approach and methods that we must follow to avoid bias in our model due to this approach. In simple terms, attenuation bias or regression dilution is the error in the model caused due to the noise in the independent variables. The noise in our model occurs due to the prediction that we use from the classification algorithm.

We also referred to one of the papers from Alibaba group titled *"Markdowns in e-commerce fresh retail: A counterfactual prediction and multi-period optimization approach"* [4] by *Junhao Hua* et. al. as they present an advanced deep learning algorithm using Convolutional Neural Networks for predicting the sales. Although we do not use this currently for our model, this is going to be one of our future works where we implement the more advanced deep learning algorithms and scale up the project from customer revenue prediction (CRP) to customer lifetime value prediction (CLV).

Data

The dataset we have worked on was released by Google Inc [8] in the year 2019 on Kaggle (it is a subsidiary of Google that provides a web interface for Data Scientists). The published dataset includes 1.71 million data points that are explained by 13 attributes.

1. fullVisitorId - This is an integer variable that is a unique value for each customer in the Google merchandise store.
2. channelGrouping – This explains the channel via which the user came to the merchandise store. The default channels set by Google Analytics [7] are used to observe the data. This is a categorical variable.
3. date – The date when the user visited the store. It is of the form 'YYYYMMDD0.'
4. device – This column consists of device information from which the user accessed the store. There are multiple attributes in this and hence the data was stored in a JSON format. The attributes that we have retrieved as separate columns are,
    - browser – The browser from which the user accessed the store
    - operatingSystem – The operating system on which the browser is running

   - deviceCategory – The type of device (Mobile, Desktop, or Tablet) they accessed the store from

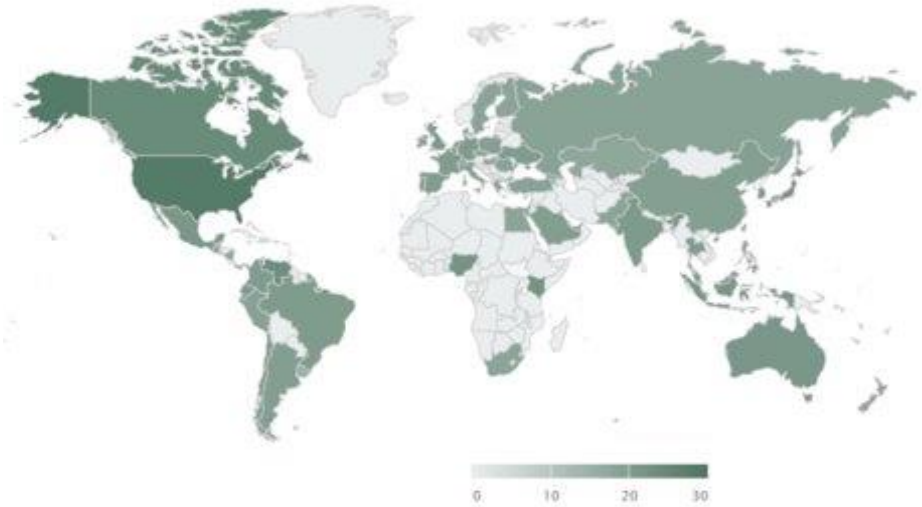
5. geoNetwork – This attribute has information about the physical location from where the user accessed the store. This also has multiple attributes within itself and is present in JSON format. The information in these attributes can be looked as a redundant but it is just more detailed information.
   - continent – The continent from where the request was sent
   - subContinent – More specific location than continent from which the request was sent
   - country – Country origin of the request
   - networkDomain – Provides information if the users queries are from the same network. All the requests from private computers/hosts in a single infrastructure are under one name
6. socialEngagementType – Explains whether the person is socially engaged or not
7. totals – This attribute is the most important attribute and holds the aggregate value of the session in JSON format. Some of the sub attributes that interest us are,
   - total_visits – The total number of visits to the site from the user in that session
   - total_bounces – Number of routers that the request has traversed through to get it the destination site
   - total_page_views – Total number of pages traversed to the destination page (Page depth)
   - total_transaction_revenue – This is going to be our “target” variable. Explains the amount of money spent during the session.
   - total_hits – Number of successful and unsuccessful hits to the website in the session
8. visitNumber – The number of sessions that the user has been active
9. visitId – This is a unique identifier for the session and usually the value stored in _utmb in the cookie folder. This is unique only for the user.
10. trafficSource – Contains information from where the traffic originated

11. visitStartTime – Timestamp of session start time in POSIX format
12. hits – This attribute stores information in a nested format (JSON) about all the page visits and hits. This is redundant data when we have other attributes.
13. customeDimensions – Information about user-level or session-level custom dimension information that is set for each session

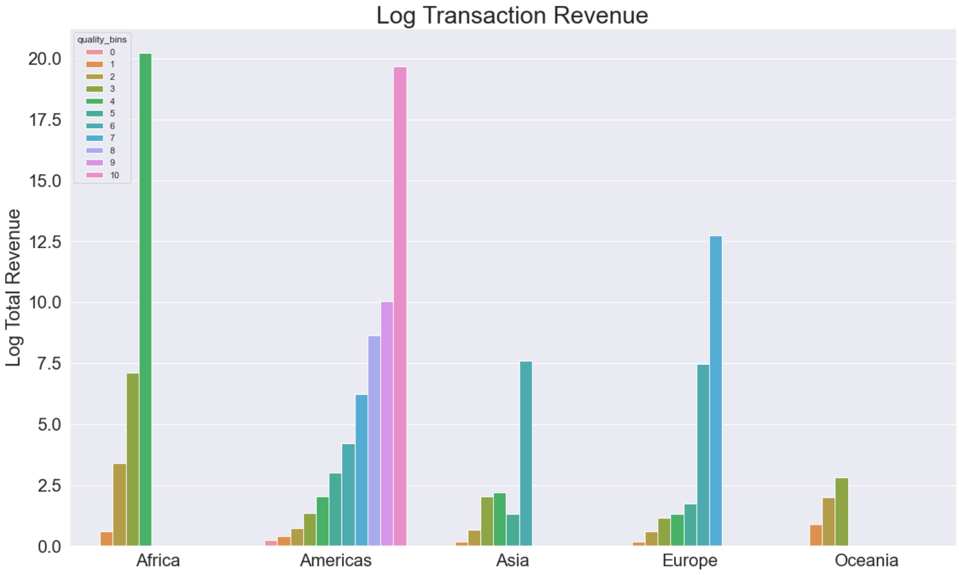
Insights

After looking closely through the dataset, we found some useful information regarding the transactions based on geographic and physical devices constraint of the user.

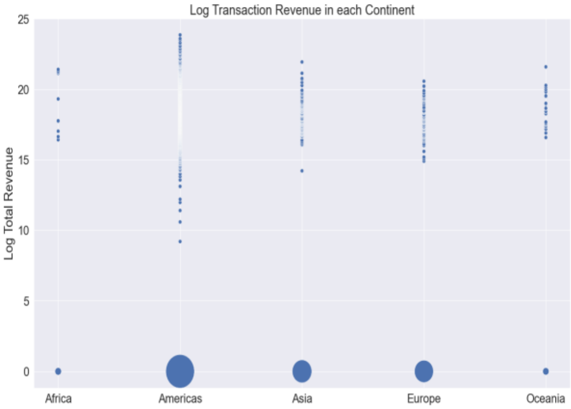
- Although the transactional revenue is spread through all the continents, North America alone has more than 85% share of all the transactions. This might be useful in categorizing and targeting the audience for specific advertisements i.e., customer segmentation.

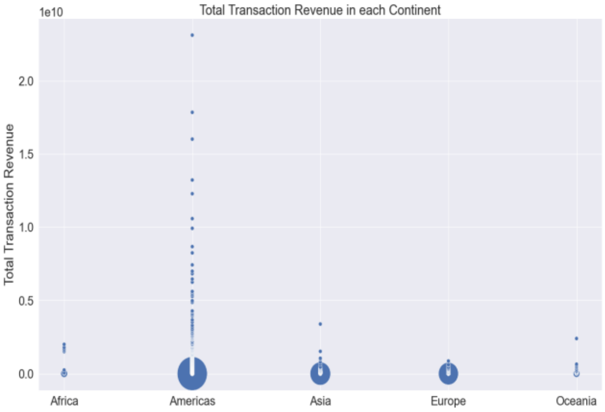


- One interesting pattern we found in the data is the relation between the session quality attribute and total transaction values. Session quality is the interaction rating score for that session. The graph below clearly shows that as the quality score increases, the transaction values are also increasing. It was surprising to see that except for America, none of the other continents has a quality score of more than 7 on a scale og 1 to 10.



- Almost 98.5% of the data has zero total transaction revenue values, and the rest 1.5% data has values in the range of millions. This is making the data highly skewed, to deal with this we applied log transformation to the total transaction revenue to decrease its variance. We can clearly see that in the two graphs below, this significantly increased our model's performance.





Methodology

Data Cleaning

We believe that Data Centric AI is the most important concept in Machine Learning, and we have worked towards it in this project. As a beginning step, we extract the important and non-redundant information from the JSON attributes and store them in separate columns. This increases our attribute count to 28. We then convert the attribute types into their relevant format and remove all the outliers in them. We then remove the columns visitId and fullVisitotId as they are unique to customers and the column socialEngagementType, as it has just one value throughout the dataset.

We then check for correlation between variables and remove the attributes total_visits and total_total_transaction_revenue as they are highly correlated with some of the other terms within the dataset. Looking at the summary statistics of all the numerical columns from the dataset, we remove outliers from the dataset. We considered the outliers to be values of anything lying outside the 3rd Standard Deviation value. We also remove the categorical column visitStartTime as it did not give any insight into the revenue value I.e., it was almost similar for all the values.

Finally, we remove the columns geo_info_network_domain, traffic_keyword, traffic_campaign, geo_info_country, and traffic_source as they are categorical variables with more than 500 categories. These many categories also did not give out information about transactional revenue.

We also convert the revenue column into a log value since the use of the logarithms of the values rather than the actual values reduces a wide range to a more manageable size as our transactional values is either 0 or very high numbers.

After all the above-mentioned data cleaning steps, we end up with a dataset of 1.69 million rows and 17 attributes to explain all these data points. We have split this dataset into training and testing with 90% of the data allotted for training and the rest 10% for testing.

Modeling

Once we are done with the whole data cleaning, we predict the customer spending value for each session. First, we predict the values directly using three regression algorithms – Linear Regression, Decision Tree Regressor, and XGBoost Regressor – and calculate the efficiency of these models using RMSE scores as shown in Table 1. To achieve a much better model, we come up with a new approach where we first predict if there is going to be a sale or not and then predict the value for the datapoint which shows that there is going to be a sale. We came up with this approach as there were nearly 98.5% of non-sale (0) values. For this task, we have used three classification algorithms – Logistic Regression, Decision Tree Classifier, and XGBoost Classifier. All the models that we have used are explained below in brief. The results are explained in Table 3.

All the models are trained on 90% of the data and tested on the rest 10%.

1. Linear Regression

The main goal of this project is to predict the actual log transactional value of the customer. For this purpose, we have built 3 regression models – Linear Regression model based on Backward Elimination, XGBoost Regressor, and Decision Tree Regressor.

Linear Regression is a simple Machine Learning algorithm of the form,

$$Y = \beta_0 + X_1\beta_1 + \cdots \dots \dots + X_p\beta_p + \varepsilon$$

where,

Y – Actual target value
X – Predictors
p – Number of predictors
β - Coefficients
ε - Error

The machine learning model tries to learn the coefficients from the predictor it sees and the output it should generate. The problem with this algorithm with this model is that it tries to fit a linear relationship between all the predictor variables and the target. With the help of these estimated coefficients, the model makes the predictions.

Backward elimination is the process of selecting predictors that are most beneficial for the model. This process is typically used when we need to reduce the number of parameters that we send into the machine learning model. The model first starts with all the predictor variables and then removes the variable with a p-value (statistical significance level) less than 0.05. Once the variables are removed, the model is trained again with the remaining attributes and this process is repeated until there are no other variables with a lower significance level. This model was used for regression purposes.

2. Logistic Regression

Logistic regression is a supervised learning method that models the probability of an event happening. This is mainly used for binary classification although it can be used for multiple by tweaking it a bit to a desirable format. The function used in this is called as the logistic function or the sigmoid function, which is of the form,

$$f(x) = \frac{e^x}{1 + e^x} = \frac{1}{1 + e^{-x}}$$

3. Decision Tree

This is a supervised machine learning model that can be used for regression and classification tasks. A tree is built incrementally by breaking the dataset into smaller and smaller subsets. The breaking of the tree is done using many different algorithms like ID3 and CART. The main concept of choosing the column to split the dataset is the amount of information that it can gain or in other terms, the split that leads to the least number of levels to reach the target/leaf node. The decision boundary of this regressor/classifier is complex I.e., an orthogonally curve boundary.

4. XGBoost

XGBoost is short for Extreme Gradient Boosting and belongs to the class of Gradient Boosting Decision Tree (GBDT) in machine learning. Gradient Boosting refers to the process of boosting or improving one weak model by combining it with several other weal models to create a more robust model. As an extension of boosting, gradient boosting, gradient boosting formalizes the process of additively creating weak models as a gradient descent method over an objective function. A set of shallow decision trees are iteratively trained by GBDTs, with each iteration using the error residuals of the prior model to fit the new model. The weighted average of all the trees is used to find the final predictions. XGBoost trees are constructed in parallel and were primarily constructed to enhance the performance and computational speeds of machine learning models. This XGBoost model is used for both regression and classification tasks.

Results

As mentioned in the methodology, we first implemented the three regression algorithms directly on the cleaned dataset. As expected, the XGBoost model is working the best among the three. Although the difference between RMSE scores is low, we should note that we are comparing the log values and hence this would be much more when calculated for the actual values. The result of all the regression algorithms is given in the below table (Table 1).

Sl. No.	Model	RMSE
1	Backward Elimination Linear Regression	1.7078
2	Decision Tree Regressor	1.669
3	XGBoost Regressor	1.623

Table 1 – RMSE scores of models when done regression directly

Now that we have the regression models working, we will go ahead and implement the new approach that we have mentioned. For that, we first build the classification algorithms, and we focus on the specificity to choose the best model. We choose the specificity as our accuracy metric as we need to reduce the number of false negatives I.e., we need to reduce the predictions of no sale when there is actually a sale as this might affect the company a lot financially. But we must also maintain the accuracy of the model as we integrate the results of this with the results of the regression model.

Sl. No.	Model	Specificity	Accuracy
1	Backward Elimination Logistic Regression	53.82	98.21
2	Decision Tree Classifier	81.15	96.78
3	XGBoost Classifier	84.19	97.15

Table 2 – Results of classification algorithms

Now that we have the classification predictions, we can integrate these results with the regression results and calculate the final RMSE. The below table shows the RMSE values for all the combinations of classification and regression algorithms. The rows tell the classification algorithm used and the columns tell the regression algorithms. The intersection of the row and column denotes the RMSE value when these algorithms were used. The best predictions that we got were for the XGBoost Classifier and the XGBoost Regressor with an RMSE value of 1.599. The RMSE score of 1.612 for a Decision Tree Classifier with an XGBoost Regressor is still better than using XGBoost alone which has an RMSE score of 1.621

	Linear Regression	Decision Tree Regressor	XGBoost Regressor
Logistic Regression	1.738	1.707	1.621
Decision Tree Classifier	1.696	1.666	**1.612**
XGBoost Classifier	1.690	1.659	**1.599**

Table 3 – RMSE scores of the Regression models when used with the classification models

Conclusion

Our approach of combining the predictions of the classification algorithm with the regression algorithm is working slightly better than the traditional machine learning algorithms. Not so surprisingly, the XGBoost Regression algorithm works the best for the above three mentioned classifiers. Although the results look like a slight improvement, the predicted results are for the log values of the revenue and hence a 0.1 change basically denotes an exponential change in the actual value.

Future Work

Although we have implemented the Customer Revenue Prediction model successfully, there is a lot of future scope for this project.

- We still have to better the accuracy and lower the RMSE using different methods
- Handle the attenuation error in a better way
- Neural networks can be implemented as the dataset looks complex
- Build a custom metric to evaluate the models

- The project can be expanded to Customer Lifetime Value prediction problem

References

(1) Optimizing Sales Forecasting in e-commerce with ARIMA and LSTM models

(2) A prediction model for imbalanced dataset using machine learning

(3) How using machine learning classification as a variable for regression leads to attenuation bias

(4) Markdowns in e-commerce fresh retail: A counterfactual prediction and multi-period optimization

(5) Predict customer lifetime value (CLV) - Dynamics 365 Customer Insights | Microsoft Learn

(6) Customer lifetime value - Wikipedia

(7) Default channel definitions - Analytics Help (google.com)

(8) Google Analytics Customer Revenue Prediction | Kaggle