GSTORE CUSTOMER REVENUE PREDICTION

Koushik Kumar Kamala
Computer Engineering
San Jose State University
San Jose, USA
koushikkumar.kamala@sjsu.edu

Madhusudhan Shagam
Computer Engineering
San Jose State University
San Jose, USA
madhusudhan.shagam@sjsu.edu

Aditi Khurd
Computer Engineering
San Jose State University
San Jose, USA
aditi.khurd@sjsu.edu

Ashna Gupta
Computer Engineering
San Jose State University
San Jose, USA
ashna.gupta@sjsu.edu

Abstract*— To analyze a Google Merchandise Store (also known as GStore, where Google swag is sold) customer dataset to predict revenue per customer. Performed data cleaning and conversion of date datatype. Then carried out feature scaling and label encoding along with cross-validation. Implemented various algorithms such as Linear Regression, Decision tree, Random Forest and a boosting algorithm Light-GBM. Best results were obtained with LightGBM. The code for this project is at *https://github.com/Madhusudhan441/GStore-Revenue-Prediction/blob/master/gstore_proj.ipynb

Keywords—Machine Learning, Revenue Prediction, Preprocessing, Linear Regression, Decision Tree, Random Forest, LightGBM.

I. INTRODUCTION

We were introduced to the problem statement through Kaggle competition. We were motivated to explore the problem and study the basic concepts of machine learning such as data preprocessing, feature scaling, label encoding, validation, and various algorithms. Basically, we analyzed a live customer dataset of GStore to learn it's revenue patterns and then made predictions for GStore revenue per customer. The predictions were based on a number of features such as channel grouping, Date of visit, Device used, Geo network, etc. These predicted values can be used to improve business strategies of companies relying on such customer data.

II. DATA

Kaggle provided us with a live dataset meaning it was supposed to get updated daily until the end of competition duration and then our results would get compared with the actual predictions made. The dataset(Figure (1)) has nine lakh rows and thirteen columns. It is a .csv file with data JSON formatted. The dataset is divided into a training set and testing set based on the dates which are a part of the dataset. Data collected till September 2018 is the training set and data collected October onwards is the test set. Training set used for evaluation has about nine lakh rows with thirteen features which after feature extraction reduced to seven lakh rows and nine features. A total of fifty-five features were extracted after decoding the JSON data. This data also had some missing values which were corrected and conversion of various datatypes was performed.

III. IMPLEMENTATION

A. Data Pre-Processing:

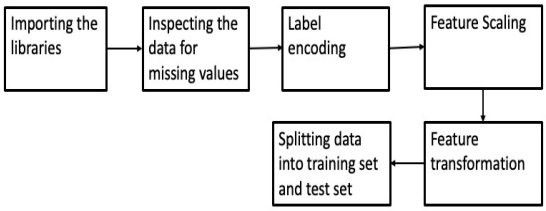

Figure (1): Data Pre-Processing Steps

In data Pre-Processing we follow the following steps

a. Import libraries
b. Import and Inspect the data
c. Label Encoding
d. Feature Scaling
e. Feature Transformation
f. Splitting the data into train and test set

Machine learning have become the most trending topic in today's technological world. Various machine learning tools exists which helps in achieving the most precise system. Data lies at the heart while dealing with such intricate system. To achieve the accuracy of the system it is the data which needs to be handled with utmost attention. Most sophisticated algorithms and machine learning tools will never make up for poor data available. It is very important to process this data before it is finally ready to be used for the machine learning model. Various data preprocessing steps are followed before it is finally ready to be used by the programmers to build a machine learning model.

a. Import Libraries:

Importing libraries is the most basic and first step in data preprocessing. It is required for running the programs used within the model. The libraries exist as a big collection of modules which are called and used further for processing. Various functions are used for using libraries. It is not required to mention each and everything in the programming world. Invoking these available functions, various things can be used without explicitly mentioning them every time within the program. In Data Science there exists popular Python libraries which can be used. Various Python libraries which exists are: Pandas, NumPy, SciPy, matplotlib, Pygame, pyQT, SymPy, SK learn, Seaborn. The libraries which is implemented in the project are: matplotlib, Pandas, NumPy, Seaborn, SK learn.

A small snippet is available for using Pandas library. It is used by assigning a shortcut "pd" in the program to use the Pandas library.

import pandas as pd

Similarly, various libraries are implemented in the code to use the required classes and functions of these libraries.

b. Import and inspect data

Datasets are present in csv format. At first the directory of csv file is necessary to be located. The data sets need to be present in the same directory as of that of your program. These are then read using read_csv function which is present in the Pandas library.

A small snippet is present to learn about the importing of dataset: pandas as pd

```
dataset = pd.read_csv(train.csv')
```

Dataset is inspected carefully. Matrix of features in the dataset(X) and their corresponding dependent vectors(Y) are created. The corresponding observations are then added accordingly.

The matrix can be developed by the proper selection of index. Proper selection of row and column index parameters i.e. row and column selection is done in order to read the columns and rows to build proper matrix. A small snippet depicting the use of iloc is shown:

```
X = dataset.iloc[:, :-1]values
```

Here we have used: to select all rows and -1 represents all columns except one.

In this manner we use various other functions of different libraries to achieve the required results. The inspection of data is performed. It is found that some part of data is missing. We have to deal with this situation, one way of dealing with this is by taking a mean of all data present in one column and replace the missing data with that value. The library which is used for this purpose is Scikit Learn preprocessing containing class called Imputer which helps to deal with the missing data.

c. Label Encoding

The data set is present in the form of qualitative form. The data can be present in textual and numerical form. When the data set is present in textual form then it becomes a bit difficult for machines to understand. It is much easier for machines to understand the numeric data and then process the given numeric data with maximum speed and best results. The textual data is present in various categories. This categorical data needs to be converted into numerical data (encoded)and the process to achieve this is called Label Encoding.

The textual data present is converted into numeric data using mathematical equations. Various calculations are involved in doing this conversion. Since the machine can understand 0 and 1 only, so the available data is then encoded in the form 0 and 1 only. Thus, encoded data is formed.The library used for doing this conversion is Sci kit library and the class we used to achieve this is LabelEncoder. The object is created for the class of LabelEncoder called labelEncoder_X. A method is also called fit_transform from LabelEncoder class itself.

It also takes into consideration two parameters X for selecting the specific number of rows and Y to select specific number of columns. Therefore, this step helps in converting the given textual categorical data into encoded form of 0 and 1. If more than two categories are present then it becomes a bit difficult to handle but can be dealt with easily after following proper methods. Numerical data is converted into floating point values using label encoding.

d. Feature scaling

In this step features are scaled in accordance to the requirement. Few features need to be removed or dealt with precisely in order to achieve the required results. Few features which are unnecessary and does not contribute in achieving a precise and accurate machine learning model can be deducted. The features which are redundant or is used more than once can also be removed. In accordance to these steps the features are transformed and required results are achieved.

e. Feature Transformation

Various algorithms are used to transform the available features. Various algorithms are implemented to transform the existing old features into new features. Various interpretations can be drawn from the new features generated and are completely different from the old features and their interpretations. As a result,

whole new set of features are created from the old set. It is applied for feature reduction. These new features generated have a different discriminatory power in different space. Some feature transformation techniques are: Scaling or normalizing, Component analyzing, using first order transformation or in accordance using second order transformation. In the project previously, we tried implementing using first order transformation, but the required results were not obtained. So later we tried using second order transformation which helped in achieving the accurate model with no faults. Second order transformation helped to achieve feature transformation using sklearn library.

f. Splitting the data into training set and test set

The data set which is available is then split into two sets which are training set and test set. Training set is very important to deal with. The features which are selected, and the data present is trained in accordance to those features. The training set is then further used and various machine learning models are applied on those training set. This helps in achieving the required system. Secondly the test set, this helps in checking the accuracy of the system. The required system which is developed needs to be checked for the accuracy.

This test set is then applied on the model developed and thus is applied to find the quality of our system. In this way the available set of data when divided into test and training set are useful. The machine learning model is responsible to learn about the correlations that exists in our training sets. After reading about these correlations we can learn about various features of the system. The test sets available are then used to check how accurately our system can predict. A general rule followed is that we must allocate 80% of data for training set and remaining 20% set of data set for testing the model.

Two libraries which are imported from libraries are test_train_split from model selection library of scikit. Four sets of data are taken into consideration. Two sets are used for testing and two other sets are used for training. X_test and Y_test related to testing part of the matrix of features and X_train and Y_train related to training part of dependent variables associated with the testing sets. *test_train_split* takes into X and Y arrays as mentioned

above and test_size as 0.5 although 0.2 was the ideal choice for selecting the data for test set.

B. Data Analysis

Various insights are achieved from the data pre-processing step. Various conclusions can be drawn from the available data present and the analysis performed on it.

After processing the data for the required g-store it was concluded that only 1.3% of users were responsible for creating non-zero revenue. For example, if there are 1000 users only 13 are responsible to create revenue rest were responsible just for visiting the store but did not contribute in the revenue generation. Secondly, it was analyzed that most of the g-store customers used chrome as their browser to shop. Most frequently operating system by the user was Windows Operating System.

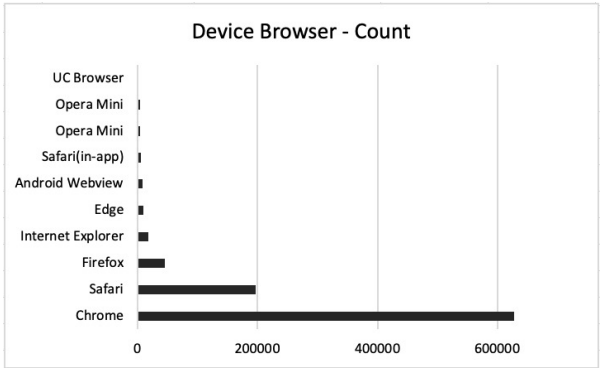

Figure (2): Analysis of Device Browser and count of users

Chrome Browser was found out to be the most frequently used browser among the G-Store users as compared to all other browsers.

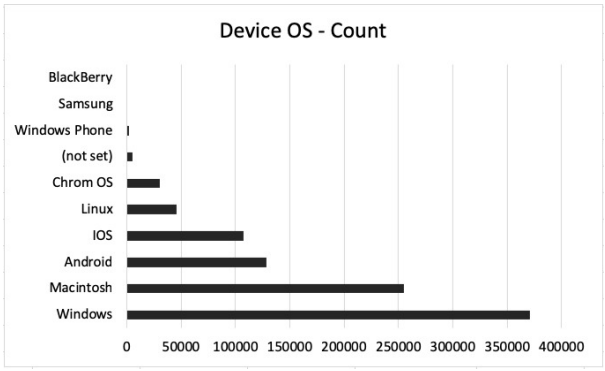

Figure (3): Analysis of Operating System used and count of users

Windows operating system is the most popular Operating system among the users. Most of the G-Store users did online shopping using this operating system.

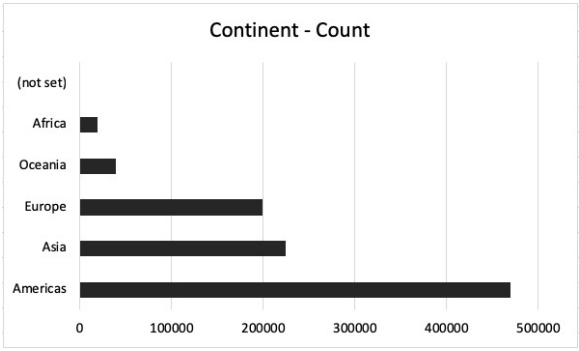

Figure (4): Analysis of Continent where users reside and count of users

It was analyzed that America and further performing more analysis of data, it was concluded that North America is responsible for creating maximum revenue of G-Store.

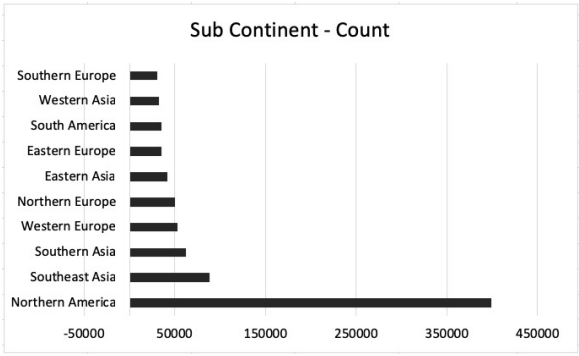

Figure (5): Analysis of Sub-Continent where users reside and count of users

These are the conclusions drawn in the form of plots. Further analyzing the data, more conclusions can be drawn. Until now we predicted the devise browser, device OS, the continent with the maximum count and the sub-continent with maximum count of G-Store users. If we try analyzing the data, more in depth then we will be able to draw more conclusions and results.

Further we have used various machine learning models. Applied these models on the processed data to achieve the required results.

IV. Applied Methods

This problem is addressed by different prediction-based learning algorithms, Linear Regression, Decision Trees, Random Forest and LightGBM.

a. Linear Regression

Linear Regression is a statistical modeling technique for finding the relationship between the target variable(Y) and the predictor variables(X), represented as $Y \approx f(X, \beta)$. This is one of the simplest, yet powerful algorithm used for Forecasting and Prediction

Given an input vector $x \in Rm$, where x1, . . ., xm represent features (also called independent variables), we find a prediction $\hat{y} \in R$ for the customer revenue $y \in R$ using a linear regression model:

$$\hat{y} = \beta_0 + x\beta_1 + \varepsilon$$

where β_0 is the intercept, $\beta 1$ is regression coefficients parameter and ε is the random error term. Here, we have trained the data using first order, second order and third order of Linear Regression and observed the results as following:

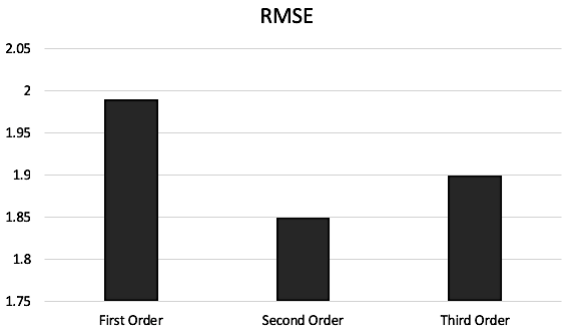

Figure (6): Root Mean Square Error for the Revenue Prediction applied at different orders of Linear Regression

The first order linear regression has resulted in RMSE of 1.99, the second order as 1.85, while with the third, the data started getting overfitted, giving RMSE of 1.9.

This model assumes that the data is independent. The learned hypothesis is fitting the training set very well but failing to generalize with new examples, thus, we have further analyzed and decided to implement that model that deals even if the nature of the dataset consists of non-linear relationships, that is applicable for continuous and categorical inputs and have implemented the decision trees.

b. Decision Tree

Decision Tree is a non-linear method, used for both regression and classification. The importance of the decision tree modeling lies in the fact that, first, decision trees are a non-parametric method and it doesn't require no preselection of variables; rather, a robust stepwise selection process is used. Thus, variables with a high clarifying power can easily be separated from the remaining, less important variables. Thirdly, regression trees easily handle both continuous and categorical variables. And if the average performance of the output classification rules is deprived, single rules can still perform extremely well and can greatly assist decision makers in improving the precision of their forecasts. This quality is mainly useful for potential customers about whom information is likely to be inadequate.

Given a set of examples of the form { 〈X, Y〉 }, where X is an n-dimensional vector X=[x1, x2, L, xn] and Y a categorical variable assuming a finite set of values (classes), a classification algorithm allows discovery of a function h(x), mapping the vector X to a class in Y:

$$h(x):X \rightarrow Y$$

where X refers to the features and the variable Y is the dependent variable. Once the function h(x) has learned from the training set, it is applied to predict the customer revenue associated with a given vector of features in test sets.

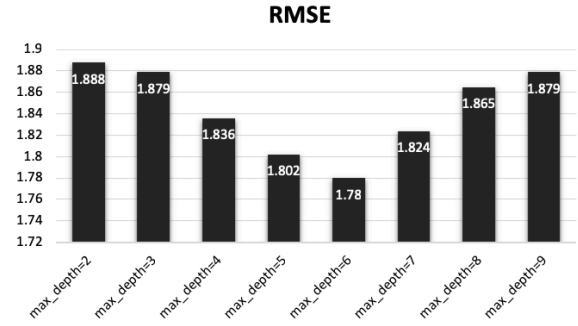

Figure (7): Root Mean Square Error for the Revenue Prediction applied at different values of max_depth of Decision Trees

This model results in RMSE of 1.78 about at a max_depth of 6, while the other RMSE are 1.88, 1.87, 1.83, 1.80, 1.82, 1.86, 1.87 for max_depth of 2, 3, 4, 5, 7, 8 and 9 respectively. As this model is prone to the problem of overfitting when it is deep, not being robust to noise and since exponentially many trees are possible due to greedy model which gives most optimal but not the global optimal, we further worked on improving the accuracy and RMSE with Ensemble methods, which are advisable for increasing large size datasets and moreover a single model alone could not fit on such a large dataset with several features. Hence, we moved towards ensemble methods.

Ensemble Methods Ensemble methods are type of meta-algorithms that combines other machine algorithms into one predictive model to decrease variance and bias. Variance can be decreased using bagging and bias can be decreased using boosting. Mainly the two types of Ensemble methods are boosting and bagging. Bagging is Bootstrap aggregation. Ideally in this method, model will train on subsets of data, which will be chosen randomly. Final results will be chosen based on voting/average of all the results. So, as we are using regression method, in our model we used average method. Boosting method is one of the popular method to convert weak learners to strong learners. In this model, weak learners are chosen for each split such as decision trees and weighted error of previous model will be applied to next model with the input. So, in boosting method, the entire dataset will be trained on series of weak models. On each stage, model input would be previous model output and error on previous model.

We tried G-Store revenue prediction on Random Forest Regressor and Light GBM model.

c. Random Forest Regression

Random Forest is the popular regressor model for predictive analytics for large dataset. It is very effective and compatible with industry level prediction problems. It is the bootstrap model, which is aggregation of decision trees. As there are many features (30) in our dataset. We applied random forest algorithm at different max_depth of decision trees and we observed best result at max_depth=9. We also tried with changing the parameters of random forest such as min_samples_split and min_samples_leaf, but we could not see any better improvement. Following are the different RMSE errors obtained by changing parameters of random forest algorithm.

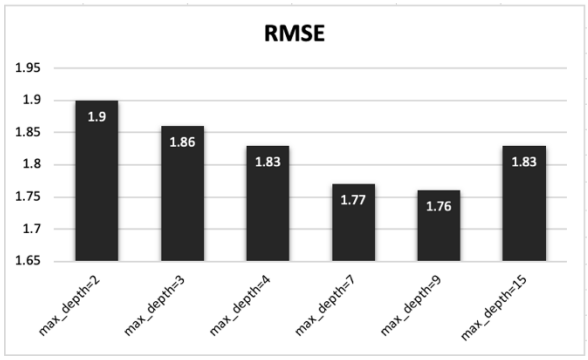

Figure (8): Random Forest results based on max_depth

As there are many features, which are both independent and dependent, we assume random forest could not fit on entire dataset as small data splits could not leverage the same output as whole output. So, after research we moved on to boosting method, which is another ensemble method.

d. LightGBM

It is a fast, high performance gradient boosting framework built on decision trees, popular for competitions for its fastness and accuracy. It is also an example of ensemble method. Based on the popular Kaggle competition stats, LightGBM shows good performance compared to other existing boosting algorithms. It works on a different formula compared to other boosting methods. Dataset will be trained on series of models. For each model, the input would be the previous model output and error occurred on previous model. We have tried with different parameters such as changing learning rate, num_leaves, min_child_samples, we have observed best result at following parameters

Num_leaves	30
Min_child_leaves	100
Learning rate	0.1
Bagging Fraction	0.7
Bagging Frequency	0.5

Table (1): Result Parameters

V. EXPERIMENTS

We tried to predict Gstore Revenue prediction on four different models Linear Regression, Decision tree, Random Forest Regressor, LightGBM boosting model. Out of all models we observed best predictions for lightGBM model with RMSE error 1.69. Below are the comparison of various models.

Models	RMSE
Linear Regression	1.85
Decision Tree	1.79
Random Forest	1.76
LGB Model	1.69

Table (2): Comparison of RMSE of different models

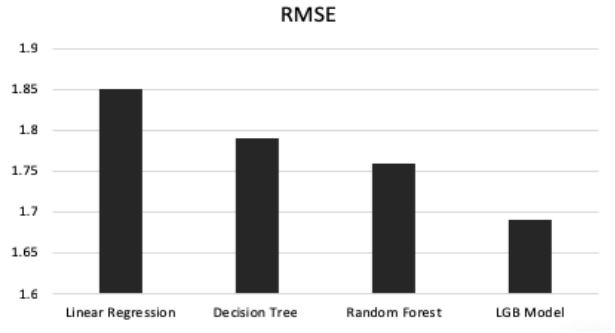

Figure (9): Visualization of various RMSE errors

VI. CONCLUSION

Our goal is GStore revenue prediction for each specific user. Ours a Kaggle competition held directly by Google Analytics group. As it is a real time competition, dataset will be updated regularly. Our dataset consists of 9 lakh rows and around 55 features. After performing data preprocessing and feature scaling, brought down to 9 lakhs by 29 features. As ours problem is a prediction model on large dataset, we choose different regressor models, which will be compatible with large dataset. We tried on Linear Regression with feature transformation but could not fit the data linearly so could not get good predictions. Similarly, with the decision tree regressor, as data is large and feature set is more, it could not map all the features into leaves. After that we tried on ensemble methods, obtained good results with boosting method using LightGBM model compared to Random Forest Algorithm. We have achieved RMSE error of 1.69 and submitted our revenue prediction for the users mentioned using the test dataset provided by Kaggle.

VII. REFERENCES

[1] Jerome Friedman, Trevor Hastie, and Robert Tibshirani: Additive logistic regression: a statistical view of boosting 2000
[2] Jerome H. Friedman: Greedy function approximation: A gradient boosting machine 2000
[3] Rubio, F. J. and Johansen, Adam M., Electronic Journal of Statistics, 2013
[4] Wang, Zhu, Electronic Journal of Statistics, 2018
[5] Chipman, Hugh A., George, Edward I., and McCulloch, Robert E., The Annals of Applied Statistics, 2010
[6] Karabatsos, George and Leisen, Fabrizio, Statistics Surveys, 2018
[7] Koltchinskii, Vladimir and Yu, Bin, The Annals of Statistics, 2004
[8] Peter C. BoxallWiktor L. Adamowicz Heterogeneous Preferences in Random Utility, 2002
[9] J. Chem. Inf. Comput. Sci. 2003, 43, 6, 1947-1958
[10] Breiman, L., Friedman, J. H., Olshen, R. A., and Stone. Classification and regression trees. Belmont, CA: Wadsworth International Group, 1984.