Walmart: Trip Type Classification

Group 1
STAT 515 Final Project 2019 Spring
Prof: Scott Bruce

Submitted by
Srashti Agrawal
Yamei Wang
Tarun Tejas Bollam

Content

1. Introduction

The purpose of this project is to classify the type of shopping trips made by customers to Walmart (*TripType*) by six predictors, including *VisitNumber*, *Weekday*, *Upc*, *ScanCount*, *DepartmentDescription*, and *FinelineNumber*, and to comparison different models' performances. The detailed description about them is shown in Table (1) below. The original dataset[1] has 647,054 observations and seven variables in total.

Table 1: Variable Description

Variable Name	**Description**
TripType	type of shopping trip the customer made
VisitNumber	single customer's ID on a single trip
Weekday	weekday of the trip
Upc	universal product number of purchased products
ScanCount	number of the given item that was purchased
Department Description	high-level description of the item's department
FinelineNumber	more refined category for each of the products

Variable Description

There are 647,054 observations. According to Table 1, there are six categorical variables: *TripType*, *VisitNumber*, *Weekday*, *Upc*, *DepartmentDescription* and *FinelineNumber*. Except for *Department Description* and *Weekday*, the rest four variables are coded as either integer or numeric variable. After making these elementary changes to the variables, data types and levels (or range for continuous variables) for these variables are provided in Table 2 below.

[1] The training set is adopted from the website since it provides response variable, *TripType*, which can be employed to calculate the accuracy rate of predictions from models. Whereas, the response variable from the test set is unavailable.

Table 2: Variable Data Types and Range		
Variable Name	**Variable Type**	**Levels or Range**
TripType	categorical	38 levels
VisitNumber	categorical	95,674 levels
Weekday	categorical	7 levels
Upc	categorical	97,714 levels
ScanCount	continuous	-12 to 71[2]
Department Description	categorical	69 levels
FinelineNumber	categorical	5,195 levels

Variable Data Types and Range

Based on Table2, it is clear that the challenge for this project is at least from two aspects: one is that there are many observations in this dataset, and the other is that many variables are categorical and they have large numbers of levels.

2. Statistical models

Three statistical models are applied to address this classification issue, and they are logistic and random forest and Gradient Boosting classifications. R code and corresponding output can be checked in Appendix I and Appendix II (separate file).

2.1 Logistic Classification

Logistic regression/maximum entropy classifier is one of the basic linear models for classification. It is usually used to predict for binary or categorical dependent variables.

According to the kernel from our reference, TripType is more associated with different items customers purchased, which is the information in three highly related variables: *Upc, FinelineNumber* and *DepartmentDescription*. Both *Upc* and *FinelineNumber* are abandoned, since they have large number of levels, so the variable *DepartmentDescription is used* as it has less(69)[3] levels to deal with. The original long format is changed into wide format by making various levels of *DepartmentDescription* as predictors, such as *'Bakery' and 'Candy'*. Thus, the new number of

[2] The negative number refers to the number of returned items.

[3] One of department description is NULL, which is renamed in my data as not_specified category. The null value are all distributed in Upc and FinelineNumber.

observation is the same as the number of customers' unique IDs, which is 95,674. VisitNumber is also abandoned in this model, because it is customer's id, and it is a unique number for each observation. The total number of purchase for each trip is calculated to give the model an extra predictor. Thus, the response variable, TripType, is predicted only by Weekday, ScanCount, TotalCount, and 69 types of *DepartmentDescription* in the logistic classification model. The original TripType with discrete numbers is not changed[4]. Weekday is also transformed into 0 to 1 binary codings. The transformed data types are displayed in Table 3 below.

Table 3: Transformed Variable Type

Variable Name	**Variable Type**	**Levels or Range**
TripType	factor	38 levels
Weekday	dummy coded	seven variables, e.g. Monday
TotalCount	int	1:209
Department Description	int	69 levels as predictors

Transformed Variable Type

The overall dataset is randomly split into the training set and a validation set with a splitting ratio 7 v.s. 3. The training set (66,971 observations) is employed to train the logistic model, the test set (28,703 observations) without response variable is used to make predictions. The function used for logistic classification is multinom( ) from nnet package. The overall accuracy of this model is 56.05%, and corresponding Log loss is 4.94.

Data exploration and Curation for random forest and Gradient boosting models:

Started with checking the data types of each variable and convert some categorical variables like *Weekday*, *DepartmentDescription*, into factors for our required output. Observations with null values are removed (4129 or less than 1% of the total observations) from the dataset because we have enough data to perform modeling. We have removed NULL department description because data is not available. After removing null values, we performed some feature generation process

[4] The different types of coding for the response variable are tried. Making the response variable into continuous integers does not necessarily improve the accuracy rate or lower the log loss score for the model. But making Triptype as a integer rather than a factor does improve the log loss score a little bit.

like creating a binary dummy variables for 7 weekdays (e.g. 1 resembles that customer purchased item on that day, 0 represent no purchase). Converting into binary dummy variable is important because if we do label coding it will take weekdays as ordinal data and we want to compare all the weekdays at equal level. Also, creation of dummy variable for each department description leading to new 68 variables, the values inside each variable translates to number of transactions made for a particular fineline number for each visit. After cleaning of data, we split it into train and test data in 70-30 ratio and used for the modeling.

2.2 Random Forest Modeling

Random forests is a supervised classification model and are a collection of decision trees. Classification is done by a 'majority vote' of the decision trees within the random forest. That is, for a given observation the class that is most frequently predicted within the random forest will be the class label for that observation. Every tree is independent of other resulting different number of trees, it is based on bagging process where all features are considered for splitting a node. The function used is random forest() from random forest package with 225 number of trees, 40 Maximum number of terminal nodes trees in the forest can have., and 60 number of variables randomly sampled as candidates at each split. After running the model the log loss value is 33.3623 which can further be optimize if we use grid search for feature selection and parametrization.

2.3 XGBoost

Gradient boosted (XGBoost) trees are a supervised learning method where a strong learner is built from a collection of decision trees in a stagewise fashion, where subsequent trees focus more on observations that were misclassified by earlier trees. The function used for gradient boosting is xgboost() from XGboost package with evaluation matrix as a log loss score. Maximum depth 15, eta is 0.05 which is used to avoid overfitting with 100 nrounds. First, we apply on the training set, getting 2.07 as a log loss score with 100 number of trees. When we apply on the test set, the log loss score is 2.25, which is quite good as compare to other classification models.

3. Model Comparisons

We use log loss value for comparing the performance of the models. Logarithmic loss (related to cross-entropy) measures the performance of a classification model where the prediction input is a probability value between 0 and 1. The range of the log loss value is from 0 to infinity. The goal is to minimize this value, log loss value increases when the predicted value diverges from the

actual value. The test log loss score defines the model accuracy. The log loss value getting from three different models are:

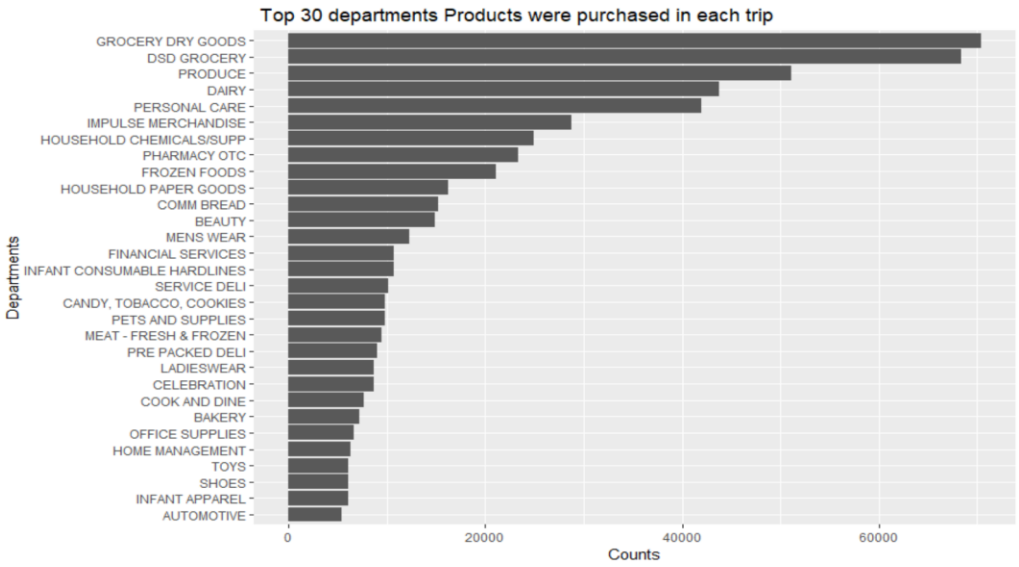
Table 4: Log loss scores	
Classification Models	**Test Log loss score**
Logistic Regression	4.94 (with different variable selection)
Random Forest	33.36 (with same variable selection)
Gradient Boosting	2.25 (with same variable selection)

Log Loss Scores

4. Visualizations

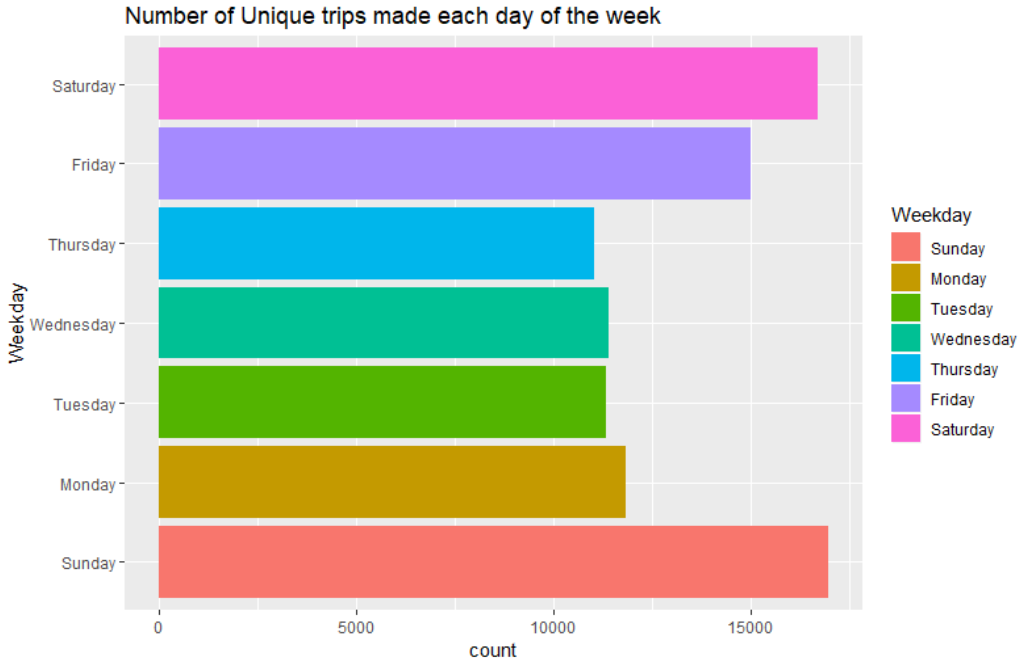
We can make some useful visualization from our data set.

1. The top 30 departments product purchased in each trip by the customer.



Plot 1. Top 30 departments frequently visited

2. Number of Unique trips made by customer each day of week.



Plot 2. Count of unique trips for each day

5. Conclusion

The prediction of triptypes to Walmart visits is very critical for the Walmart development. It can facilitate goods arrangement and staff deployment to meet customers' requirements for single visit. Different machine learning models are applied in this prediction project. The key challenge for this project is feature engineering, which is directly associated with the performance of each model. Thus, two different sets of variables are selected for logistic model and random forest, XGBoost models to maximize the dataset usage. The results of model comparisons show that Gradient Boosting classifier is superior as compare to Random Forest, and logistic Regression has in the middle between them but as we used different set of variables we cannot compare to other two models in terms of prediction performances but we can be consider variable generation feature as it has low log loss value. Hence, the lowest log loss is for gradient boosting so, it is recommended to use XGBoost for accurately predicting trip types to Walmart made by customers.

For further study, it is recommended that different feature engineering can be employed before being applied to different machine learning models. Also, deep learning models can be applied to this for large dataset. A well-designed network with sufficient hidden layers might improve performance accuracy or lower Log loss score.

References

https://www.kaggle.com/c/walmart-recruiting-trip-type-classification/data

https://stackoverflow.com/questions/35013822

https://rdrr.io/cran/MLmetrics/man/LogLoss.html

en.wikipedia.org/wiki/Gradient_boosting