**DATASCI 450**

**Class Project | Group 10 | FINAL REPORT**

# **Project Title**

***Walmart Recruiting: Trip Type Classification***

# **Team Member**

***Aleksey Kramer, Jagger Bodas, Winston Featherly-Bean***

**TABLE OF Contents**

[**Project Title** 1](#_Toc437271933)

[**Team Member** 1](#_Toc437271934)

[**Project Description** 2](#_Toc437271935)

[**Objective and Scope** 2](#_Toc437271936)

[**PROJECT: Part 1** 3](#_Toc437271937)

[**Exploratory Data Analysis and Data Issue Identification (Questions a and b)** 3](#_Toc437271938)

[**TOTAL NUMBER OF RECORDS** 3](#_Toc437271939)

[**SUMMARY STATISTICS FOR EACH COLUMN** 3](#_Toc437271940)

[**OBSERVATIONS** 6](#_Toc437271941)

[**HIGH LEVEL ANALYTIC PROBLEM (Question C)** 6](#_Toc437271942)

[**PROJECT: Part 2** 7](#_Toc437271943)

[**PROJECT: Part 3** 8](#_Toc437271944)

[**Summary: Part 1, analysis 8**](#_Toc437271945)

[**Summary: Part 2, model building 9**](#_Toc437271946)

[**Next steps 13**](#_Toc437271947)

[**REFERENCES / ATTACHMENTS** 13](#_Toc437271948)

[**PROJECT Part 1 Attachments** 13](#_Toc437271949)

[**Project Part 2 Attachments** 13](#_Toc437271950)

# **Project Description**

Walmart uses both art and science to continually make progress on their core mission of better understanding and serving their customers. One way Walmart is able to improve customers' shopping experiences is by segmenting their store visits into different trip types.

Whether they're on a last minute run for new puppy supplies or leisurely making their way through a weekly grocery list, classifying trip types enables Walmart to create the best shopping experience for every customer.

Currently, Walmart's trip types are created from a combination of existing customer insights ("art") and purchase history data ("science"). In their third recruiting competition, Walmart is challenging Kagglers to focus on the (data) science and classify customer trips using only a transactional dataset of the items they've purchased. Improving the science behind trip type classification will help Walmart refine their segmentation process.

# **Objective and Scope**

For this competition, we are tasked with categorizing shopping trip types based on the items that customers purchased. To give a few hypothetical examples of trip types: a customer may make a small daily dinner trip, a weekly large grocery trip, a trip to buy gifts for an upcoming holiday, or a seasonal trip to buy clothes.

Walmart has categorized the trips contained in this data into 38 distinct types using a proprietary method applied to an extended set of data. We are challenged to recreate this categorization/clustering with a more limited set of features. This could provide new and more robust ways to categorize trips.

The training set (train.csv) contains a large number of customer visits with the TripType included. You must predict the TripType for each customer visit in the test set (test.csv). Each visit may only have one TripType. You will not be provided with more information than what is given in the data (e.g. what the TripTypes represent or more product information).

# **PROJECT: Part 1**

“Define the objective and scope of the project. Gather and organize data for the project.

1. Conduct exploratory data analysis such as visualizing the data through graphs, tables, summary statistics, and other means to understand the data.
2. Identify any issues associated with data gap, data size, data type, data manipulation, data storage, and data retrieval for analysis. Structured or unstructured data?
3. Describe the high level analytic problem needs to be resolved: supervised learning, unsupervised learning.”

## **Exploratory Data Analysis and Data Issue Identification (Questions a and b)**

### **TOTAL NUMBER OF RECORDS**



### **SUMMARY STATISTICS FOR EACH COLUMN**

|  |
| --- |
| **TripType** |
|  |

|  |  |
| --- | --- |
| **VisitNumber** | |
|  |  |

|  |  |
| --- | --- |
| **WeekDay** | |
|  |  |

|  |
| --- |
| **UPC** |
|  |

|  |
| --- |
| **ScanCount** |
|  |

|  |  |
| --- | --- |
| **Department Description** | |
|  |  |

|  |
| --- |
| **FileLineNumber** |
|  |

### **OBSERVATIONS**

We observe missing values for UPC, FileLineNumber. Which cannot be replaced. And hence we decided to drop incomplete rows. So we are left with 642925 rows of data. Since it’s structured data we are storing it in an R data frame data structure.

We are planning to undertake following potential feature engineering tasks

* Add field for weekend / workday
* Create Shopping cart category (small / medium / large )
* Categories Departments
* Identify entropy of shopping cart
* Create Shopper Profiling - we can do clustering to determine profiling
* Define Boolean value - returns / no returns
* Do Basket Analysis modelling to get additional features.

## **HIGH LEVEL ANALYTIC PROBLEM (Question C)**

As we are tasked with categorizing shopping trip types based on the items that customers purchased and we also have labeled historical data , this problem could be solved using multi-class classification techniques which is a supervised learning technique..

# **PROJECT: Part 2**

“Part 2: Model construction and evaluation

a)      Construct analytic model(s) to address the project objective

b)      Evaluate the model outcomes

c)       Iterate and improve the model when necessary

d)      Justify the final model and its output”

During the second phase of the project team members were attempting to build different models to produce the highest accuracy. Feature engineering yielded several additional dimensions added to the data set.

Aleksey was trying to build a tree adding and testing dimensions to the data set, but get caught up in mangling with data more than concentrating on the actual model. The model Aleksey produced roughly yields 48% accuracy, which is significantly lower than the model that was built by Jagger.

Jagger worked to converted Department Description column to multiple binary columns. As well as worked to create new columns such as basket\_item\_count, basket\_return\_count , basket\_size. And then merged rows corresponding to each basket / Visit into single row , so that this data could be used for classification.

Jagger utilized Random Forest algorithm to produce a model that is almost 66% accurate. Jagger used ensembling approach to test accuracy of the different types of models. Random Forest approach yielded best results. As such, the Random Forest model produced by Jagger is a clear winner. In addition, Jagger also ranked the model following Kaggle’s guidelines – Jagger script contains the details. Our team received a Kaggle Score of 1.65507.

Aleksey’s and Jagger’s scripts can be found in the appendix portion of this document.

Winston is working on an ensemble model that offers a "majority rules" prediction based on the outputs of Jagger's random forest model, a support vector machine model, and an XGBoost model. The support vector machine model, on its own, achieved 70% precision in testing.

We will be also exploring ways to extract / create new useful features from Upc and FinelineNumber columns to further improve our accuracy.



# **PROJECT: Part 3**

This final section summarizes the work done in the first two parts, reports on our performance in the Kaggle competition, and sets out our plans for future iterations.

## Summary: Part 1, analysis

Our first step was to understand the classification problem posed, and to assess the data. We were tasked with categorizing shopping trips into one of 38 different “trip types” – or, more precisely, for a given shopping trip, stating with what probability it belongs to each of those categories.

We identified this as a multiclass classification problem with categorical and numerical inputs, best solved with supervised machine learning techniques.

The provided training dataset had 647,054 rows and 7 columns; each row represented an item purchase (or return), and included an integer value to indicate the trip type. Trips in which the customer purchased multiple items were represented by multiple rows in the dataset. About four thousand rows were missing values; we deleted these.

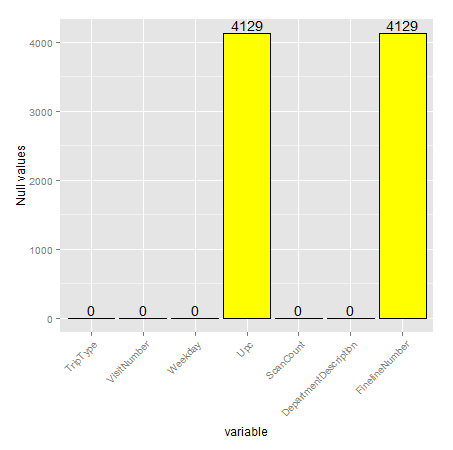


Figure - Features in original dataset, with number of null values

To implement our SML algorithms, the dataset needed to be reshaped such that each row represented a single trip. This would require, at minimum, the creation of new columns (features) to represent each of the possible items or item categories. Other potential engineered features included a weekend indicator, the size of the basket of items and the diversity of the basket.

## Summary: Part 2, model building

In the second phase, Jagger and Aleksey munged and reshaped the data, while all three team members individually built and tested SML models. Background reading on previous Kaggle competitions suggested that a successful strategy would likely entail combining several different models. Our individual attempts at model-building provided a diversity of approaches for our eventual submissions. We worked in R.

**Project workflow**

Input

Submission

Output

Ensemble selection

Support vector machine

Rpart classification tree

Random forest

Weekends

Purchases / returns count

Basket size

Merge same-trip records

Drop incomplete records

Kaggle / Walmart dataset

Feature engineering

Preprocessing

Aleksey created a sqlite database for quicker data exploration. Jagger converted the “department description” feature into multiple columns, added features related to weekends, basket sizes and total purchases/returns, and merged all rows from the same visit. This reshaped dataset was the training data for our models.

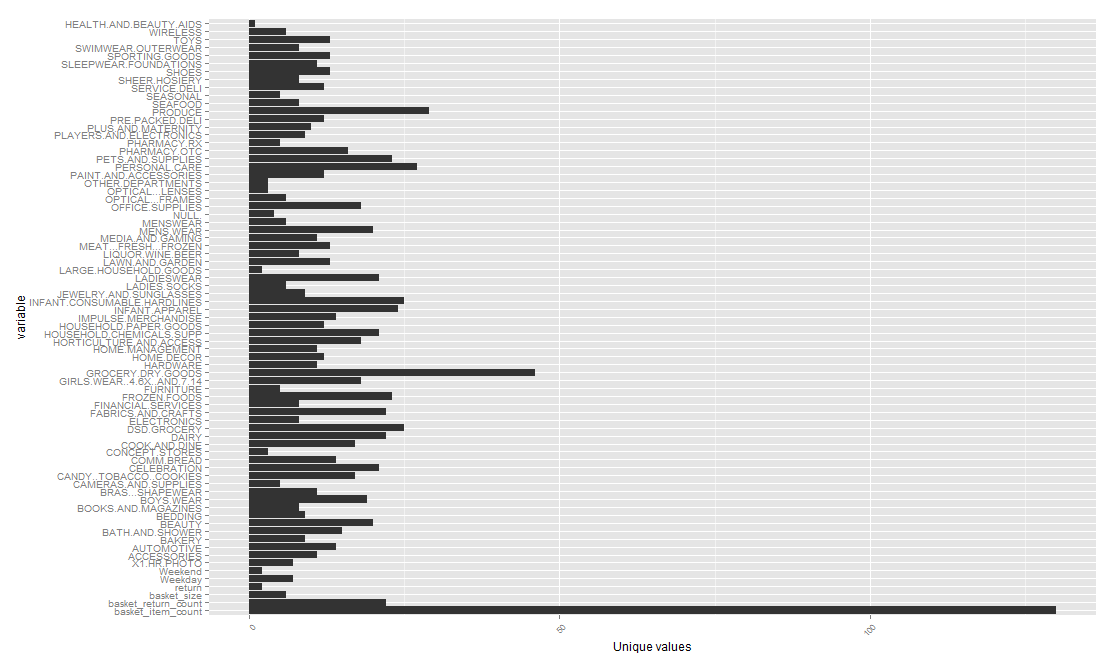


Figure - Reshaped dataset, with # of unique values per feature

Between us, we ultimately used three models: a Random Forest model (“randomForest” library for R), an implementation of the CART family classification tree (from the “rpart” library), and a support vector machine model (from the “e1071” library).

Our first submission was the Random Forest model alone. The Kaggle competition scores entries using a multi-class logarithmic loss metric[[1]](#footnote-1); this submission scored 1.65507, placing 361st as of 7 December 2015. A second submission combined predictions of the three models through a “majority rules” ensemble model: each base model predicted a single label for each trip, then the ensemble model predicted the label chosen by a majority of models (or, in a tie, the label chosen by the higher-performing SVM model[[2]](#footnote-2)). This submission fared poorly – scoring 11.61, or 564th place as of 7 December – as it predicted a single label for each record, rather than the distribution of probabilities encouraged by the competition’s framework and scoring metric.

For the third submission, we predicted probabilities for each of a shopping trip’s possible labels, then again combined these via an ensemble model that output a weighted average of those predicted probabilities. This also scored disappointingly: 1.89680, much worse than either the random forest or support vector machine models alone.

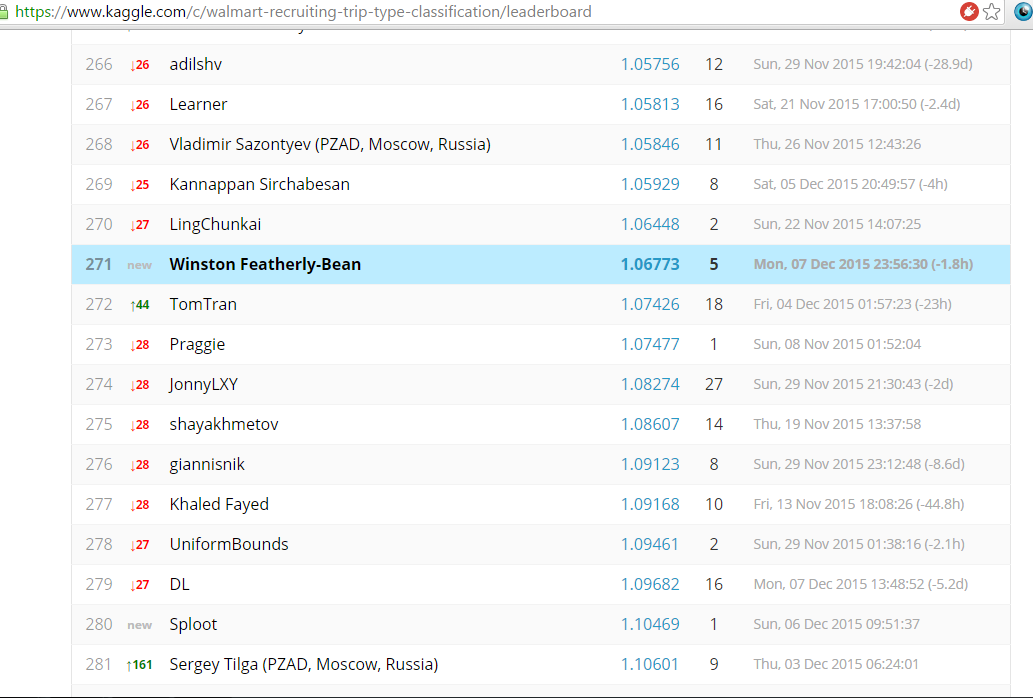


Figure - Our team's best score came from the SVM model trained on the features specified above

## Next steps

Time permitting, our team has several ideas for improving our performance. These include:

**Tuning the SVM model’s hyperparameters**: this model’s performance is known to vary widely depending on hyperparameter (gamma, c) values, with no way of knowing the optimal parameters a priori. A common approach is a grid search, iteratively building and comparing SVM models over a range of possible hyperparameter values, then selecting the best performer. Attempts to do this in Part 2 failed, as the R implementation did not finish running even after multiple days. Given more time, or a more optimized implementation, we expect to see a performance gain from this step.

**Trying new features**: we used the least granular item category available as an input. This had benefits in terms of speed, but may cost predictive power. We also have ideas for engineered features still to be implemented, including shopper cluster membership and shopping basket diversity measures. These steps might also require some data engineering, e.g. using sparse matrices, to keep processing times down.

**Introducing new model types and fine-tuning their weighting**: ensemble approaches tend to improve as more diverse models are included. We can build other sorts of models, e.g. multinomial logistic regression or XGBoost, and include them in the averaged predictions. Playing with the weights assigned to each model’s predictions – typically giving more weight to models with higher standalone performance – could also lead to gains.

# Things we tried and didn’t work out (ran out of time)

During Part 2 analysis we had excluded Upc and FinelineNumber columns from feature Engineering exercise. And hence we started exploring how to include features related to these 2 columns.

First approach Jagger took to create Association rules exclusively corresponding to each of these 2 columns and then filter rules with High confidence (>0.5). And only include columns featuring in above filtered rules. However as Association rule does not include Scancount information or duplicate items in basket , we decided to not pursue it further.

Then Jagger compared frequencies of unique values in these 2 columns and realized any Upc or FinelineNumber with Frequency more than 50 is worth considering as that forms majority of dataset. So in case of FinelineNumber Unique values to be considered came down from 5196 to 2135. And in case of Upc unique values to be considered came down from 97714 to 1537.

Jagger wrote R code to convert Upc , FinelineNumber from long format to wide format i.e. create column for each of the selected Upc/FinelineNumber and record ScanCount value corresponding to Item.

With above changes now we had 3750 columns in our training data set after feature engineering. And we tried to apply RandomForest , SVM in R but due to resource constraints those model training exercises could not finish. Hence Jagger decided to try Azure ML so that we will not have any resource constraint. Due to modular code R code we were able to port our logic quickly in Azure and start experimenting. Unfortunately we could not finish our Azure ML experimentation before submission deadline (as Azure ML experiment is still running) , hence we will still to our v1 version and output. You can see revised feature engineering code for v2 version below in attachments.

# **REFERENCES / ATTACHMENTS**

## **PROJECT Part 1 Attachments**

1. EXPLORATORY DATA ANALYSIS R SCRIPTS (TEAM) 
2. EXPLORATORY DATA ANALYSIS - OUTPUT IN PDF FORMAT (TEAM) 

## **Project Part 2 Attachments**

1. Classification attempt using rpart package
2. Random forest implementation
3. Data Files: 

## **Project Part 3 Attachement**



1. “The formula is:

   −1N∑i=1N∑j=1Myijlog(pij),

   where N is the number of visits in the test set, M is the number of trip types, log is the natural logarithm, yij is 1 if observation i is of class j and 0 otherwise, and pij is the predicted probability that observation i belongs to class j.”

   (<https://www.kaggle.com/c/walmart-recruiting-trip-type-classification/details/evaluation>, accessed 7 December 2015) [↑](#footnote-ref-1)
2. This scored 1.06773 on the logarithmic loss metric on its own. [↑](#footnote-ref-2)