Classification Challenge

“Core” Mission

- Progress through understanding and serving the customer.

- **Source of current classification**
  - Using customer insights a.k.a. “ART”
  - History data a.k.a. “SCIENCE”

- **Problem Statement**
  - To classify customer trips using transactional dataset of items purchased.
  - Segmenting store visits into different trip types.

Walmart

Why……….?

- **……….? (Obvious Opinion)**
  - To refine segmentation Process
  - The "Core" Mission!!

- **……….? (Oblivious POV)**
  - To gauge any possible change in shopping motivations.
  - Effective promo display.
  - Product placement and assortment.
  - Effect of store's layout on type of customers.

Data fields

- **TripType** - The ground truth that we are predicting, a categorical id representing the type of shopping trip the customer made.
- **VisitNumber** - an id corresponding to a single trip by a single customer
- **Weekday** - the weekday of the trip
- **Upc** - the UPC number of the product purchased
- **ScanCount** - the number of the given item that was purchased.
- **DepartmentDescription** - a high-level description of the item's department
- **FinelineNumber** - a more refined category for each of the products, created by Walmart

Deliverables

- Treatment of data: missing values and outliers
- Exploratory analysis of data
- Feature engineering
- Application of supervised learning algorithm
  1. XGBoost
  2. Randomforest
  3. Gradient boosting machine
- Submission in Kaggle.

Data Transformation

- Handling NULL values
  - Out of a total of 647054 rows, 4129 rows have NULL values (less than 1%)
  - Assuming data is missing at random, ignore rows with NULL values

- ‘Weekday’ field converted to binary (whether the day of visit is a weekend or not)
  - If the day is Friday, Saturday or Sunday – it is considered as weekend (i.e. value of the field is 1)
  - Else it is weekday

- Negative values in ‘Scancount’
  - Indicates a return of the item.
  - Return of an item does not affect buying pattern
  - ‘Scancount’ is made 0 for negative values

Reshaping Data

- Raw Data: Each item bought by a customer at every visit

	TripType	VisitNumber	Weekday	Upc	ScanCount	DepartmentDescription	FinelineNumber
1	999	5	Friday	68113152929	-1	FINANCIAL SERVICES	1000
2	30	7	Friday	60538815980	1	SHOES	8931
3	30	7	Friday	7410811099	1	PERSONAL CARE	4504
4	26	8	Friday	2238403510	2	PAINT AND ACCESSORIES	3565
5	26	8	Friday	2006613744	2	PAINT AND ACCESSORIES	1017
6	26	8	Friday	2006618783	2	PAINT AND ACCESSORIES	1017
7	26	8	Friday	2006613743	1	PAINT AND ACCESSORIES	1017
8	26	8	Friday	7004802737	1	PAINT AND ACCESSORIES	2802
9	26	8	Friday	2238495318	1	PAINT AND ACCESSORIES	4501
10	26	8	Friday	2238400200	-1	PAINT AND ACCESSORIES	3565
11	26	8	Friday	5200010239	1	DSD GROCERY	4606
12	26	8	Friday	88679300501	2	PAINT AND ACCESSORIES	3504
13	26	8	Friday	22006000000	1	MEAT - FRESH & FROZEN	6009
14	26	8	Friday	2236760452	1	PAINT AND ACCESSORIES	7
15	26	8	Friday	88679300501	-1	PAINT AND ACCESSORIES	3504
16	26	8	Friday	2238400200	2	PAINT AND ACCESSORIES	3565

Using 'dcast' function aggregate data such that a row represents the number of each item purchased by a customer in a particular visit.

TripType	VisitNumber	Weekday	1-HR PHOTO	ACCESSORIES	AUTOMOTIVE	BAKERY	BATH AND SHOWER	BEAUTY	BEDDING
35	154673	Monday	0	0	0	0	0	0	0
35	154699	Monday	0	0	0	0	0	0	0
35	154706	Monday	0	0	0	0	0	0	0
35	154875	Monday	0	0	0	0	0	0	0
35	154901	Monday	0	0	0	0	0	2	0
35	154955	Monday	0	0	0	0	0	0	0
35	155114	Monday	0	0	0	0	0	0	0
35	155361	Monday	0	0	0	0	0	1	0
35	155378	Monday	0	0	0	0	0	0	0
35	155542	Monday	0	0	0	0	0	0	0
35	155599	Monday	0	0	0	0	0	0	0
35	155648	Monday	0	0	0	0	0	0	0
35	155796	Monday	0	0	0	0	0	0	0
35	155827	Tuesday	0	0	0	0	0	0	0

Normalise the rows so that row sum of all products is 1.

Feature Correlation Graph

- Compute correlation matrix of reshaped training data
- Compute adjacency matrix from the correlation graphs as follows:-
  - If absolute value of correlation is less than a threshold (0.05 in this case), assume there is no correlation between the purchase of items and value in adjacency matrix is 0 i.e. there is no edge between these 2 products in the correlation graph.
  - Otherwise value in adjacency matrix is 1 i.e. there is an edge between the products in the graph.
  - All diagonal elements in the adjacency matrix are made 0 to avoid self loops.

Correlation Matrix

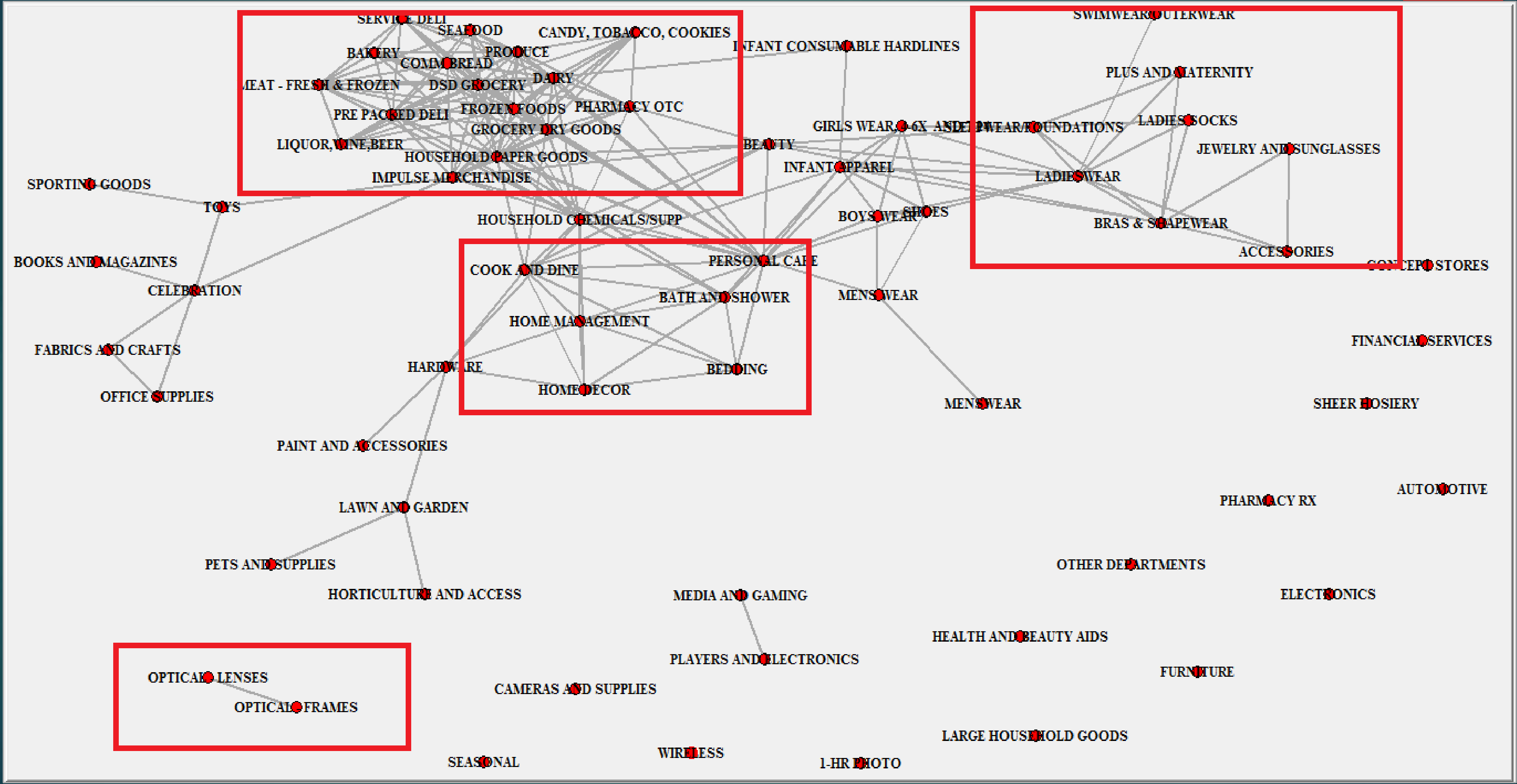
	1-HR PHOTO	ACCESSORIES	AUTOMOTIVE	BAKERY	BATH AND SHOWER	BEAUTY	BEDDING
1-HR PHOTO	1.00000	-0.00281	-0.00510	-0.00691	-0.00257	-0.00250	-0.00340
ACCESSORIES	-0.00281	1.00000	-0.00542	0.00173	0.02114	0.03776	0.00189
AUTOMOTIVE	-0.00510	-0.00542	1.00000	-0.00802	-0.00214	0.00429	-0.00230
BAKERY	-0.00691	0.00173	-0.00802	1.00000	0.00046	0.00322	0.00328
BATH AND SHOWER	-0.00257	0.02114	-0.00214	0.00046	1.00000	0.04085	0.21161
BEAUTY	-0.00250	0.03776	0.00429	0.00322	0.04085	1.00000	0.01970
BEDDING	-0.00340	0.00189	-0.00230	0.00328	0.21161	0.01970	1.00000

Adjacency Matrix

	1-HR PHOTO	ACCESSORIES	AUTOMOTIVE	BAKERY	BATH AND SHOWER	BEAUTY	BEDDING
1-HR PHOTO	0	0	0	0	0	0	0
ACCESSORIES	0	0	0	0	0	0	0
AUTOMOTIVE	0	0	0	0	0	0	0
BAKERY	0	0	0	0	0	0	0
BATH AND SHOWER	0	0	0	0	0	0	1
BEAUTY	0	0	0	0	0	0	0
BEDDING	0	0	0	0	1	0	0

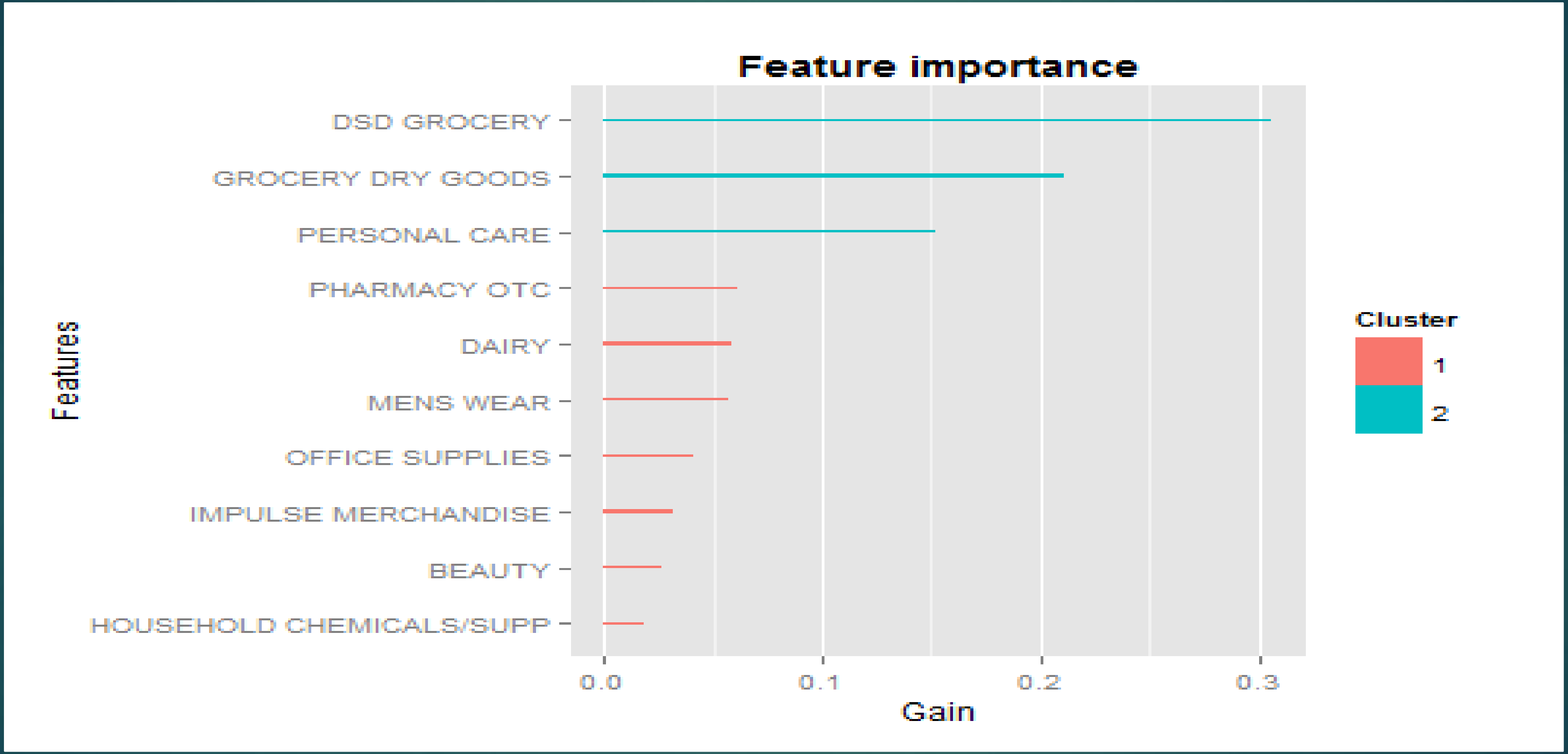
Walmart

Feature Correlation Graph



Walmart

Feature Importance Graph



XGBoost

XGBoost is short for extreme Gradient Boosting. It is

- An open-sourced tool – Computation in C++, R interface provided
- A variant of the gradient boosting machine – Tree based model
- The winning model for several Kaggle competitions

Basic Walkthrough

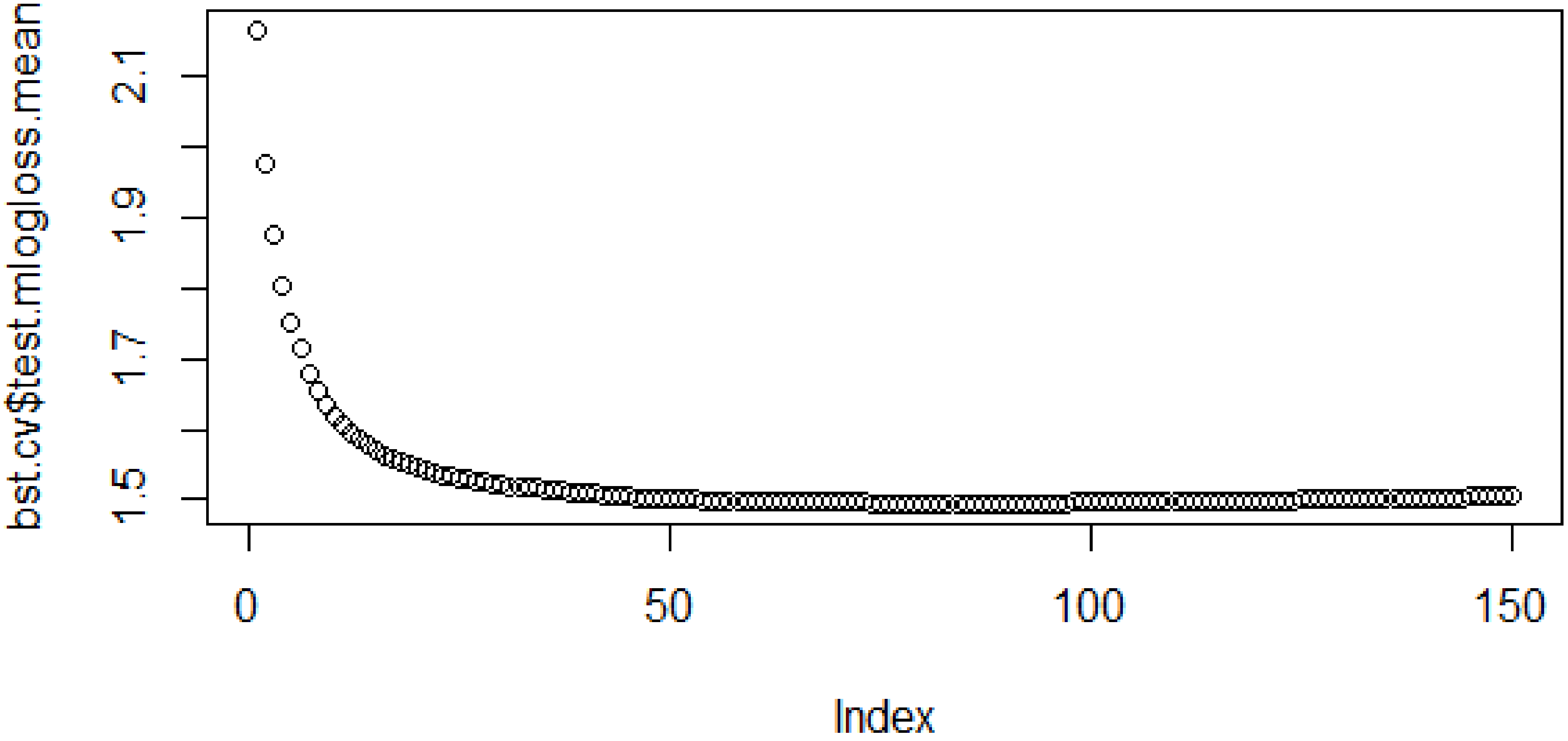
- The algorithm works only on numeric matrices, hence we need to preprocess the data.

```r
#keep record of the test id for final output

id = test[,1]

#remove the id column

train = train[,-1]
test = test[,-1]

#convert the target from character into integer starting from 0

target = train$target
classnames = unique(target)
target = as.integer(colsplit(target,'_',names=c('x1','x2'))[,2])-1

#remove the target the from train

train = train[,-ncol(train)]

#convert dataset into numeric Matrix format

trainMatrix <- data.matrix(train)
testMatrix <- data.matrix(test)
```

Cross-validation and model building

- Once the data has been reshaped into the required format, we can choose cross validation to find to choose the parameters.
- **numberOfClasses:** is equal to 38, since there are 38 classes in total
- **param**: parameters of the model with “objective” indicating the task, “eval_metric” indicating the error measurement of the model
- **cv.nround**: number of the trees to build. This is the parameter we want to tune
- **cv.nfold**: how many parts you want to divide the train data into for the cross-validation
- **bst.cv**: run the cross-validation

```r
#cross-validation to choose the parameters

numberOfClasses <- max(target) + 1

param <- list("objective" = "multi:softprob",
              "eval_metric" = "mlogloss",
              "num_class" = numberOfClasses)

cv.nround <- 500
cv.nfold <- 5
bst.cv = xgb.cv(param=param, data = trainMatrix, label = target,
                nfold = cv.nfold, nrounds = cv.nround)

plot(bst.cv$test.mlogloss.mean,lty = 'l')
nround <- which(bst.cv$test.mlogloss.mean==min(bst.cv$test.mlogloss.mean))

#train the model

bst = xgboost(data = trainMatrix, label = target, param=param, nrounds = nround)

#predict the model

ypred = predict(bst, testMatrix)
```


bst.cv$test.mlogloss.mean
2.1
1.9
1.7
1.5
0
50
100
150
Index

Performance evaluation Metric

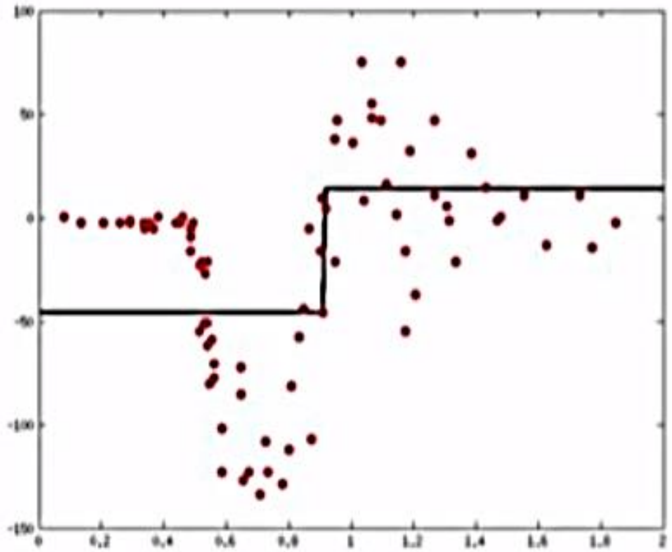
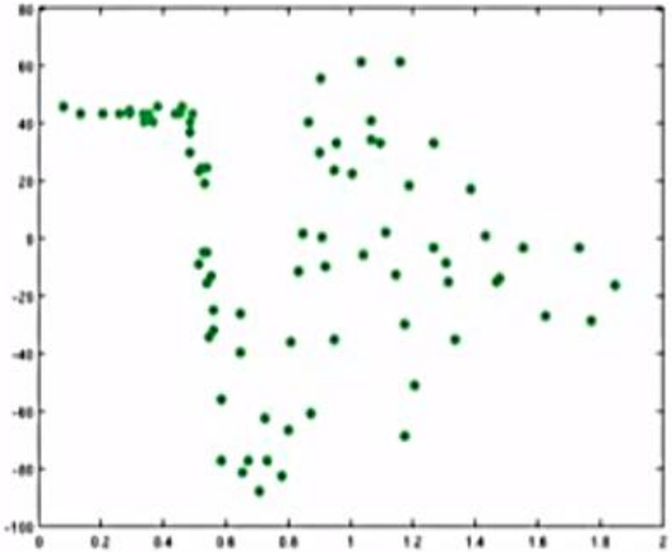
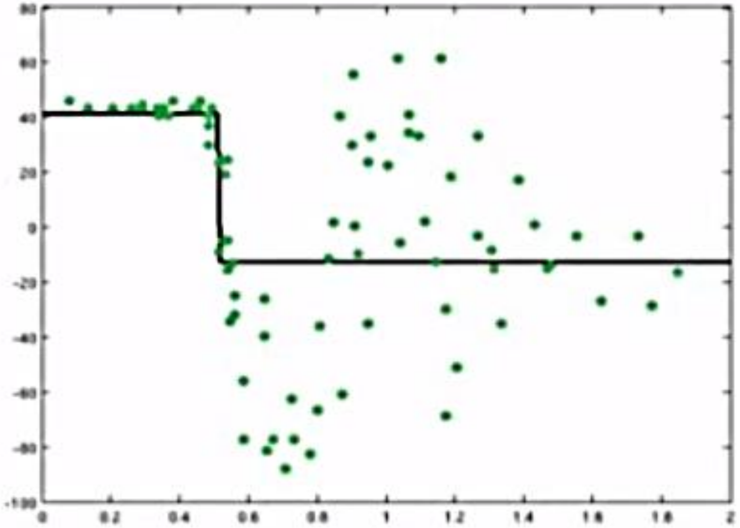
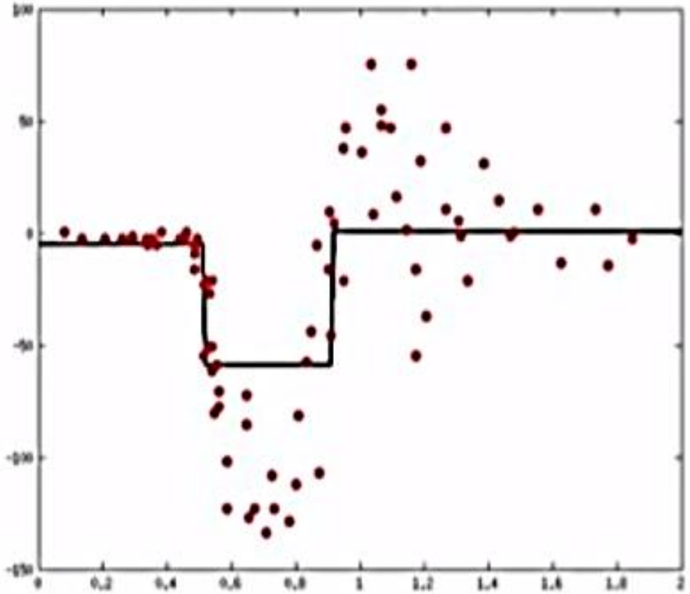
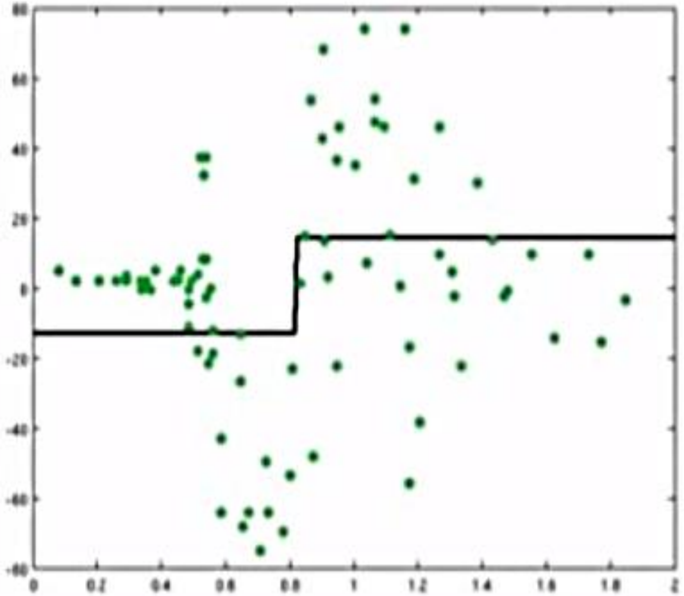
- Logloss function $-\frac{1}{N} * \sum_{i=1}^{N} \sum_{j=1}^{M} y_{ij} \log(p_{ij})$
- N is the number of visit in the test set.
- M is the number of trip types.
- y_{ij} is 1 if observation ‘i’ belongs to class ‘j’ and 0 otherwise.
- p_{ij} is the predicted probability.

Bagging : Random Forest

- Ensemble of decision trees.
- Unlike single decision trees, Random Forests use averaging to find a natural balance between the two extremes.
- Random forest uses bootstrapping and averaging.
- Out of bag error estimate by using department description as features is 44.5%
- This implies department Description alone is not a good classifier.

Boosting: Gradient Boosting Machine

- Fit complex models by iteratively fitting sub-models (decision tree) to residuals.
- Gradient boosting uses a "pseudo gradient"
- Pseudo-gradient used is the derivative of a general loss function L().
- In this case: logloss-function.
- It shows the deviation of predicted probability of class from original training example.
- A sub-learner is picked as close as possible to the pseudo gradient and added to model.

Challenges and Bottlenecks

- **Memory issues**: With limited RAM, handling big numeric matrix was not feasible.
- dcast() function is not useful in reshaping features ~5K
- Different number features in test data and train data when features are made using   FineLinenumber and departmentDescription.
- Department description is not enough for classification.
- No improvement even after trying different classification algorithms

Results

Mon, 30 Nov 2015 19:47:04 3rd submission Edit description	submission.zip	2.15648	☐
Mon, 30 Nov 2015 19:35:02 2nd submission Edit description	submission.zip	2.28584	☐
Fri, 13 Nov 2015 23:01:58 1st Submission. Edit description	submission.zip	1.72548	☐

Submit Selection Changes

Walmart

THANK YOU!