Using Decision Tree to predict repeat customers

Jia En Nicholette Li (niclje@stanford.edu) Jing Rong Lim (jingrong@stanford.edu)

Motivation

A study by Bain & Company stated 25% to 95% increase in profits can be made by increasing 5% of customer retention rates and a 30% rise in company value with an increase of 10% of customer retention[1]. As Decision Tree algorithms inherently estimate the suitability of features during separation of classes and can handle both categorical and numerical features, this project aims to derive the ranking of features in retaining customers with the use of product offers.

Aim

With an input of **number of items bought, total amount spent in 30 days/ 60 days/ 90 days/ 180 days/ overall transactions** from a **product company/ product category/ product brand**, find out the usefulness of **Decision Tree algorithm** to **predict repeat customers** and **determine the important features for predicting repeat customers**.

Methodology

Data

The dataset is acquired from the Kaggle competition, *Acquire Valued Shoppers Challenge* containing data with these properties.

Data Type	Properties
Past Transactions	Customer ID, store, product department, product company, product category, product brand, date of purchase, product size, product size, product measure, purchase quantity, purchase amount
Training History	Customer ID, store, offer ID, geographical region, number of repeat trips, repeater, offer date
Testing History	Customer ID, store, offer ID, geographical region, number of repeat trips, repeater, offer date
Offers	Offer ID, offer category, offer quantity, offer company, offer value, offer brand

It contains nearly 350 million rows of past customers' pre-offer transactions but we are using 15.4 million rows to make it more manageable for this project. The transactional data was reduced based on the criteria that the transaction category was a category on at least one of the offers.

Feature Engineering

Due to the nature of the data (a single customer buying many different products), it was necessary to merge the transactions from each customer into a row. After which, the pre-offer transactions was processed and the following set of features for each customer were engineered:

For every individual company, category and brand -

1. Total quantities bought
2. Total amount spent
3. Total amount spent within 30 days before offer date
4. Total amount spent within 60 days before offer date
5. Total amount spent within 90 days before offer date
6. Total amount spent within 180 days before offer date
7. Never bought from particular company/ category/ brand

Implementing Decision Trees

Our model was implemented using *Apache Spark*'s machine learning library (v1.5.1) that uses distributed CART (Classification And Regression Trees) to construct the tree based on numerical splitting criterion recursively applied to the training data.

Basic Algorithm

The decision tree is a greedy algorithm that performs a recursive binary partitioning of the feature space by selecting the best split from a set of possible splits to maximize the information gain at a tree node from the set: $\{ \underset{s}{\operatorname{argmax}} \; IG(D,s) \}$

Node impurity and information gain

The *node impurity* is a measure of the homogeneity of the labels at the node. The current implementation provides two impurity measures for classification, Gini impurity and entropy.

Impurity	Formula
Gini Impurity	$\sum_{i=1}^{C} f_i (1 - f_i)$
Entropy	$\sum_{i=1}^{C} -f_i \log(f_i)$

Results & Discussion

Classification

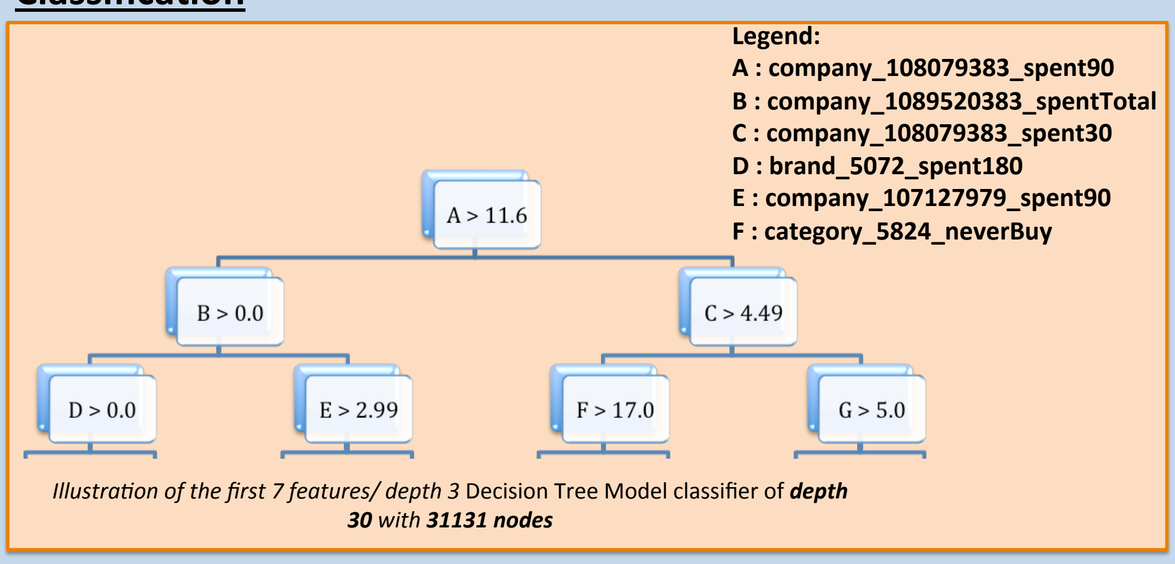

Illustration of the first 7 features/ depth 3 Decision Tree Model classifier of ***depth 30*** *with* ***31131 nodes***

The above Decision Tree model was built using **Entropy** impurity measure, with a maximum depth of 30 - the maximum depth this library allows. The decision to select the maximum depth possible was made after taking into careful consideration the amount of features the training dataset contains (close to 400), and it is thus unlikely for the Decision Tree model to overfit the data. When training the model, a 70/30 split on the data was implemented to perform cross validation, which attain a test error of < 5%.

Feature Selection

Decision Trees implicitly applies feature selection whilst performing classification. When fitting a decision tree to training data, the top few nodes of which the tree is split are regarded as the important variables within the dataset and feature selection is thus completed automatically as the features are sorted top down by information gain. We used the heuristic, Separability of Split Value (SSV) criterion[2], for feature selection as one of the basic advantage of using SSV is that it can be applied to both continuous and discrete features, as well as compare the estimates of separability despite the substantial difference in the types of data.

The top few nodes of the tree are regarded as the most important variables in feature selection, and features that appeared infrequently are pruned, thus simplifying the Decision Tree model. In this case, 112 out of 399 features had not appeared at all and 58 features have a frequency of less than or equal to 10 occurrences. These are pruned from the Decision Tree and can be ignored when performing classification.

Top 10 recurring features:

Feature label	Count
company_104127141_itemsTotal	12468
company_104127141_spentTotal	9504
company_106414464_neverBuy	3062
company_104127141_spent30	2904
company_1076211171_neverBuy	2788
company_104127141_neverBuy	2528
company_106414464_itemsTotal	2234
company_106414464_spentTotal	1862
company_107106878_spentTotal	1666
company_106414464_spent30	1598

Conclusion

The Decision Tree model is a good model to predict customer repeats and it achieved a test error of <5% upon cross validation. As an added benefit of using Decision Trees, we came out with the top 10 recurring features and noticed that a company (id: 104127141) is a dominant feature in predicting whether a customer repeats.

References

[1] Reichheld, F. (2001). *Prescription for cutting costs.* Bain & Company. Boston: Harvard Business School Publishing.

[2] K. Grabczewski and W. Duch. A general purpose separability criterion for classification systems. In Proceedings of the 4th Conference on Neural Networks and Their Applications, pages 203–208, Zakopane, Poland, June 1999.

[3] K. Grabczewski and Norbert Jankowski. Feature Selection with Decision Tree Criterion. Dept. of Comput. Methods, Nicolaus Copernicus Univ., Torun, Poland, Nov 2005