Predicting RedHat Business Value

Chirag Jain
Department of Computer Science
cj63715@uga.edu

Sharmin Pathan
Department of Computer Science
sp59764@uga.edu

Abstract

Every organization records behaviors and activities of the individuals that interact with it (customers). Customers are the most essential part of any business. It is important to identify and measure customer loyalty and increase revenue. The customers' data gathered acts as a key point of differentiation that helps to attract new customers in competitive business environments. This data helps the organization to identify its potential customers and create ways to approach them. RedHat maintains this information over time to study its customers' behavior. We propose a solution to identify which customers have a potential business value for RedHat based on their activities and its characteristics.

1 Overview

[2] RedHat is world's leading provider of open source, enterprise IT solutions. Today, they deliver a comprehensive portfolio of secure products and services using the same open, collaborative business model and an affordable, predictable subscription model. Since 1993, RedHat has been helping its customers navigate through technology and business disruption though open innovation. RedHat promotes free and open source software, methodology, and culture in education. They have been at the forefront of open source for more than 20 years.

Like most companies, RedHat is able to gather a great deal of information over time about the behavior of individuals who interact with them. They're in search of better methods of using this behavioral data to predict which individuals they should approach—and even when and how to approach them. With an improved prediction model in place, RedHat will be able to more efficiently prioritize resources to generate more business and better serve their customers.

The data science question addressed here is to classify customer potential. We propose to create a prediction model that will help to more efficiently prioritize resources and generate more business and better serve the customers. For each activity, a probability of the outcome is to be computed. The outcome indicates whether each person has completed the activity within a fixed window of time after it was performed. The challenge is to predict the potential business value of a person who has performed a specific activity. We would suggest better methods of using this behavioral data to predict which individuals need to be approached – and even when and how to approach them. (This was a competition hosted on Kaggle)

1.2 Problem Statement

To create a classification algorithm that accurately identifies which customers have the most potential business value for RedHat based on their characteristics and activities. This is a binary classification problem.

2 Preliminaries

The system was operational with the following libraries used for feature selection and building a classification model.

2.1 Technologies Used

Recursive Feature Elimination (RFE) [3]: The data features used to train machine learning models have a huge influence on the performance that the model can achieve. Irrelevant or partially relevant features can negatively impact model performance. Having irrelevant features in the data can considerably decrease the accuracy.

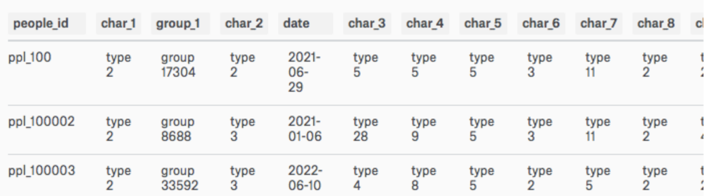
RFE uses model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute. It basically gives the feature ranking. It provides an external estimator that assigns weights to features (e.g. coefficients of a linear model), the goal of RFE is to select features by recursively considering smaller and smaller sets of features. First, the estimator is trained on an initial set of features and weights are assigned to each one of them. Then, features whose absolute weights are the smallest are pruned from current set features. That procedure is recursively repeated on pruned set until desired number of features to select is eventually reached.

Random Forest Classifier [4]: Random forest is an ensemble learning method for classification and regression. It operates by constructing multiple decision trees during training and predicts the class labels (classification) based on the individual trees. A random forest is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and use averaging to improve the predictive accuracy and control over-fitting. The sub-sample size is always the same as the original input sample size but the samples are drawn with replacement if bootstrap=True (default).

3 Dataset

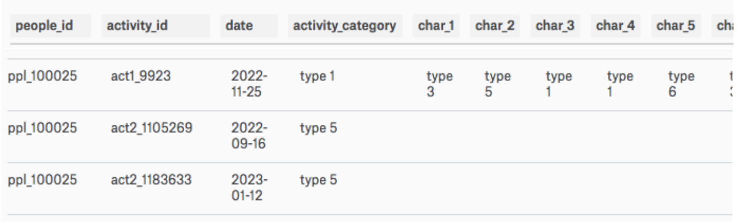
[1] The dataset was taken from https://www.kaggle.com/c/predicting-red-hat-business-value/data. It includes people.csv, act_train.csv and act_test.csv. The people file contains all the unique people (and the corresponding characteristics) that have performed activities over time. Each row in the people file represents a unique person. Each person has a unique people_id. The activity files contain all the unique activities (and the corresponding activity characteristics) that each person has performed over time. Each row in the activity file represents a unique activity performed by a person on a certain date. Each activity has a unique activity_id. The activity file is comprised of several activities with differing characteristics. Type 1 activities are different from type 2-7 activities as there are more known characteristics associated with type 1 activities than type 2-7 activities that have only one associated characteristic. The act_train.csv set contains 2,197,291 rows and 15 columns. people.csv has 189,118 rows and 41 columns, and act_test has 498,688 rows and 14 columns (no outcome field in the test set). people.csv spans upto 38 characteristics. char_1 to char_9 are string values, char_10 to char_37 are boolean values, and char_38 contains numerical values.

The challenge is to predict potential business value of a person who has performed an activity over time. This business value in terms of the activity outcome is a yes/no field attached to every activity in the activity file. The outcome defines whether a person completed the activity within a given time period after the activity was performed.



people_id	char_1	group_1	char_2	date	char_3	char_4	char_5	char_6	char_7	char_8
ppl_100	type 2	group 17304	type 2	2021-06-29	type 5	type 5	type 5	type 3	type 11	type 2
ppl_100002	type 2	group 8688	type 3	2021-01-06	type 28	type 9	type 5	type 3	type 11	type 2
ppl_100003	type 2	group 33592	type 3	2022-06-10	type 4	type 8	type 5	type 2	type 5	type 2

Figure 1. people.csv (columns span upto char_38)



people_id	activity_id	date	activity_category	char_1	char_2	char_3	char_4	char_5	ch
ppl_100025	act1_9923	2022-11-25	type 1	type 3	type 5	type 1	type 1	type 6	t
ppl_100025	act2_1105269	2022-09-16	type 5						
ppl_100025	act2_1183633	2023-01-12	type 5						

Figure 2. act_train.csv

3.1 Exploratory Data Analysis

Several graphs were plotted to analyze the data and summarize the main characteristics.

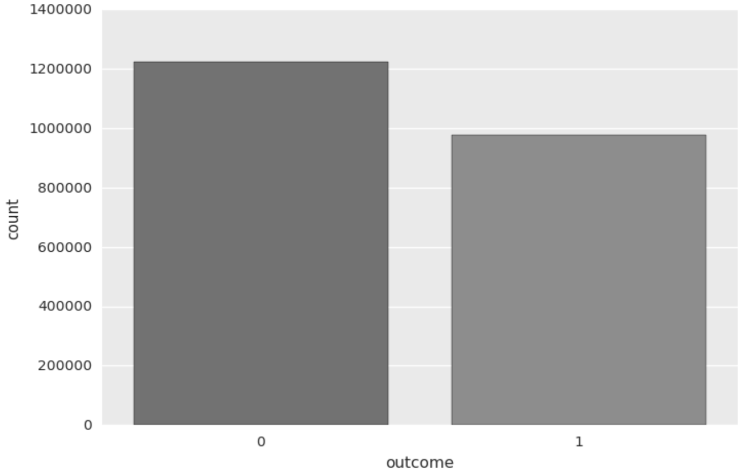


Figure 3. total record count vs outcome

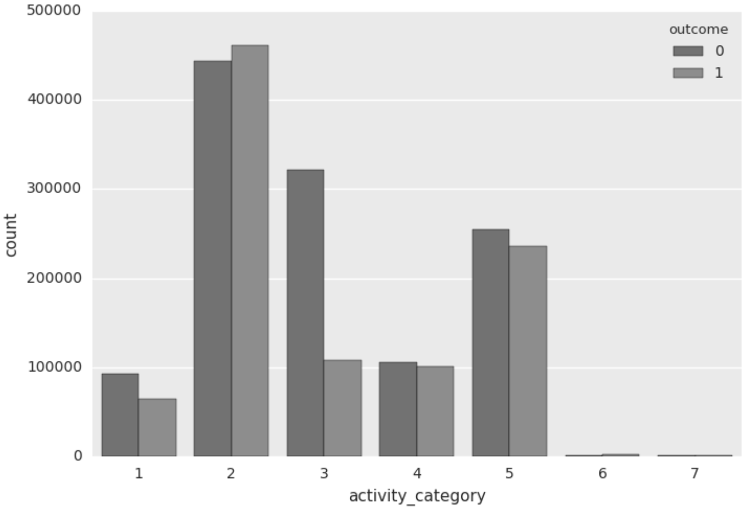


Figure 4. total record count vs activity_category

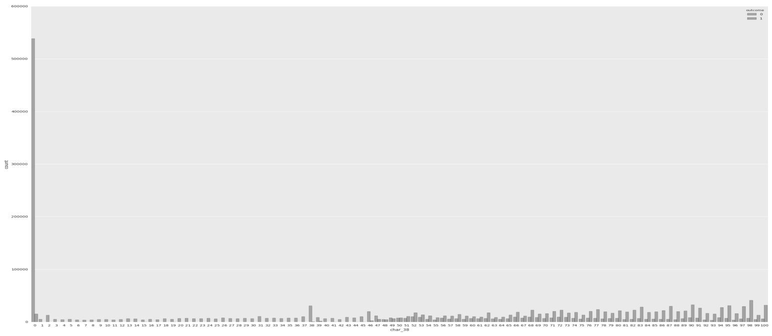


Figure 5. total record count vs char_38

4 Approach

This section speaks of the preprocessing and model building strategies.

4.1 Preprocessing

The categorical and boolean attributes from people and activity files have been converted to numerical attributes. The date column was broken into date, month, and year attributes. The date column was then dropped. This unification to numerical fields was done so that the attributes can contribute towards model building. There was a suggestion on kaggle to merge people and activity files into a single dataset to build a predictive model. This merging was performed using the people_id attribute which was common in both the files. After merging, '-1' was filled in for missing values.

people_id and activity_id columns were dropped since they are not informative about the outcome status. Finally, the data and labels were separated to be fed to the model.

4.2. Feature selection

Recursive Feature Elimination (RFE) was used as a feature selection approach. RFE works by recursively removing attributes and building a model on those attributes that remain.

RFE output: Column Headers (['char_1_x', 'char_2_x', 'char_3_x', 'char_4_x', 'char_5_x', 'char_6_x', 'char_7_x', 'char_8_x', 'char_9_x', 'char_10_x', 'year_x', 'month_x', 'day_x', 'char_1_y', 'group_1', 'char_2_y','char_3_y', 'char_4_y', 'char_5_y', 'char_6_y', 'char_7_y','char_8_y', 'char_9_y', 'char_10_y', 'char_11', 'char_12','char_13', 'char_14', 'char_15', 'char_16', 'char_17', 'char_18','char_19', 'char_20', 'char_21', 'char_22', 'char_23', 'char_24','char_25', 'char_26', 'char_27', 'char_28', 'char_29', 'char_30','char_31', 'char_32', 'char_33', 'char_34', 'char_35', 'char_36', 'char_37', 'char_38', 'year_y', 'month_y', 'day_y'])

Feature Ranking ([22, 23, 20, 18, 19, 6, 7, 16, 17, 28, 14, 4, 26, 1, 27, 1, 9, 8, 3, 1, 15, 10, 11, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 24, 1, 21, 1, 1, 1, 1, 1, 5, 2, 13, 12, 25])

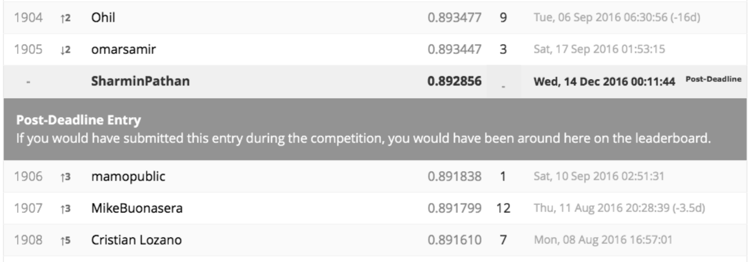
The attributes with lowest ranking were less informative about the outcome, these were removed before preparing the data for model building. The attributes with the least 10 rankings were removed from the training and testing sets.

4.3. Building a Predictive Model

Training was performed using Random Forest Classifier as an evaluating algorithm. 46 attributes with the highest rankings returned by RFE were fed to the model. 4-fold cross validation was applied reserving 3-folds for the training data and 1-fold for the validation data. The outcomes were then predicted for the testing data set. A submission file containing the activity_id and corresponding outcomes was prepared. Random Forest classifier was built around 96 trees.

5 Performance

A submission was made on kaggle.com which got us an accuracy of 89.2856



1904	↑2	Ohil	0.893477	9	Tue, 06 Sep 2016 06:30:56 (-16d)	
1905	↓2	omarsamir	0.893447	3	Sat, 17 Sep 2016 01:53:15	
-		**SharminPathan**	**0.892856**	-	**Wed, 14 Dec 2016 00:11:44**	**Post-Deadline**

Post-Deadline Entry
If you would have submitted this entry during the competition, you would have been around here on the leaderboard.

1906	↑3	mamopublic	0.891838	1	Sat, 10 Sep 2016 02:51:31
1907	↑3	MikeBuonasera	0.891799	12	Thu, 11 Aug 2016 20:28:39 (-3.5d)
1908	↑5	Cristian Lozano	0.891610	7	Mon, 08 Aug 2016 16:57:01

Figure 6. Submission on Kaggle

5.1 Evaluation Metric

The evaluation metric used by kaggle for this challenge was area under the ROC curve between the predicted and the observed outcome.

[5] Receiver operating characteristic (ROC) or ROC curve is a graphical plot which is an excellent way to measure the performance of a binary classifier system. In ROC curve, the true positive rate (TPR) is plotted against the false positive rate (FPR) at various threshold settings. The true-positive rate(TPR) is also known as sensitivity, recall or probability of detection in machine learning. The false-positive rate(FPR) is also known as the fall-out or probability of false alarm and can be calculated as (1 − specificity). The area under the ROC curve (AUC) is a measure of how well a parameter can distinguish between two outcomes (yes/no). An area under the curve of value 1.0 represents a perfect classification; and of 0.5 and below represents a worse classification; between 0.5 and 1.0 represents good classification.

Considering our two-class prediction problem (binary classification), in which the outcomes are labeled either as positive (1) or negative (0). There are four possible outcomes from a binary classifier. If the outcome from a prediction is '1' and the actual value is also '1', then it is called a true positive (TP); however, if the actual value is '0' then it is said to be a false positive (FP). Conversely, a true negative (TN) has occurred when both the prediction outcome and the actual value are '0', and false negative (FN) is when the prediction outcome is '0' while the actual value is '0'.

True Positive Rate = True Positive/ (True Positive + False Negative)
False Positive Rate = False Positive/(False Positive + True Negative)

Our ROC score: 0.8883361

6 Concluding Remarks

We performed RFE and Random Forest classification for predicting the outcome of the activities performed by RedHat's customers. The results obtained however can be greatly further improved on. A different feature selection strategy or tuning the parameters of Random Forest can help get a better accuracy. Customers are crucial to the success of any business as they are the fundamental source of revenue. Identifying and valuing the potential customers helps the organization advance. As the number of entries and data continue to grow rapidly, a faster mode of processing is the need of the hour hence we could work towards optimization to produce faster and more accurate results.

References

[1] Predicting RedHat Business Value | Kaggle
https://www.kaggle.com/c/predicting-red-hat-business-value

[2] RedHat
https://www.redhat.com/en/about/company

[3] scikit-learn Feature Selection
http://scikit-learn.org/stable/modules/feature_selection.html

[4] scikit-learn Random Forest Classifier
http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

[5] Receiver Operating Characteristic Wiki
https://en.wikipedia.org/wiki/Receiver_operating_characteristic