Capstone 2 Project Final Report

Predicting Red Hat Business Value - Classify Customer Potential

Enpei Xi

1. Background

Like most companies, Red Hat 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.

In this project, the challenge is to create a classification algorithm that accurately identifies which customers have the most potential business value for Red Hat based on their characteristics and activities.

With an improved prediction model in place, Red Hat will be able to more efficiently prioritize resources to generate more business and better serve their customers.

2. Potential clients

The potential client in this project is definitely the Red Hat. With the data Red Hat, a predictive model can be built based on the features in this data set. This model can be used for predicting potential business of a person who has performed a specific activity in future cases.

Other companies may use this model as well. With similar data architecture, the model could predict for any classification problem in business. What it needs is to retrain the model to fit the new data architecture.

3. Datasets Description, Exploration and Wrangling

The data set is coming from Kaggle Competition - 'Predicting Red Hat Business Value'. This competition uses two separate data files that may be joined together to create a single, unified data table: a people file and an activity file.

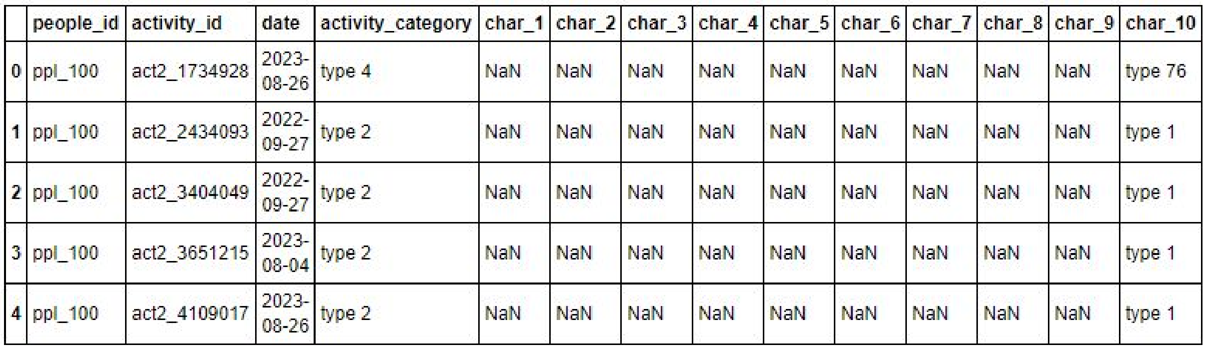
3.1 Data Description

The people file contains all of 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 file contains all of 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.

Each categorical features contain several categorical variables, and NaN values as well. What I need to do is to modify them all into one same category, and we can say that category to be an 'undefined group', and they are good to fit into the model to be trained.

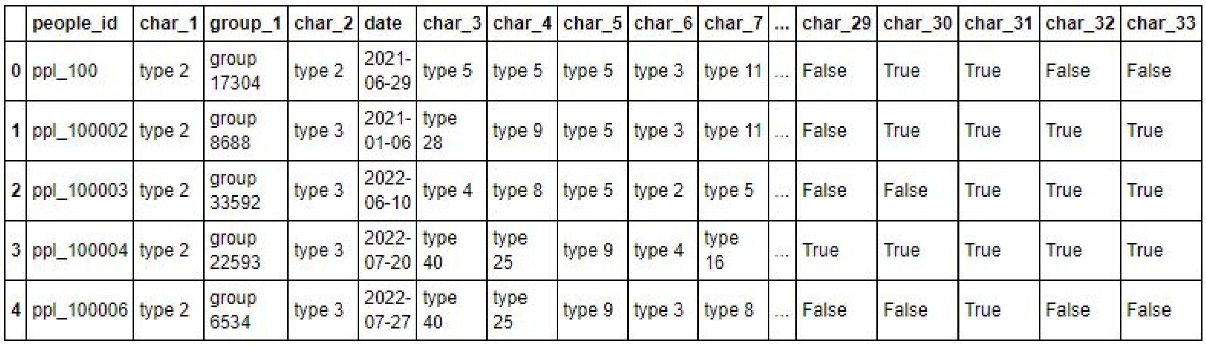
The activity file



	people_id	activity_id	date	activity_category	char_1	char_2	char_3	char_4	char_5	char_6	char_7	char_8	char_9	char_10
0	ppl_100	act2_1734928	2023-08-26	type 4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 76
1	ppl_100	act2_2434093	2022-09-27	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1
2	ppl_100	act2_3404049	2022-09-27	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1
3	ppl_100	act2_3651215	2023-08-04	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1
4	ppl_100	act2_4109017	2023-08-26	type 2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	type 1

- The activity file contains several different categories of activities. As the data description of the competition, Type 1 activities are different from type 2-7 activities because there are more known characteristics associated with type 1 activities (nine in total) than type 2-7 activities (which have only one associated characteristic).
- The business value outcome is defined by a yes/no field attached to each unique activity in the activity file. The outcome field indicates whether or not each person has completed the outcome within a fixed window of time after each unique activity was performed.

The people file



	people_id	char_1	group_1	char_2	date	char_3	char_4	char_5	char_6	char_7	...	char_29	char_30	char_31	char_32	char_33
0	ppl_100	type 2	group 17304	type 2	2021-06-29	type 5	type 5	type 5	type 3	type 11	...	False	True	True	False	False
1	ppl_100002	type 2	group 8688	type 3	2021-01-06	type 28	type 9	type 5	type 3	type 11	...	False	True	True	True	True
2	ppl_100003	type 2	group 33592	type 3	2022-06-10	type 4	type 8	type 5	type 2	type 5	...	False	False	True	True	True
3	ppl_100004	type 2	group 22593	type 3	2022-07-20	type 40	type 25	type 9	type 4	type 16	...	True	True	True	True	True
4	ppl_100006	type 2	group 6534	type 3	2022-07-27	type 40	type 25	type 9	type 3	type 8	...	False	False	True	False	False

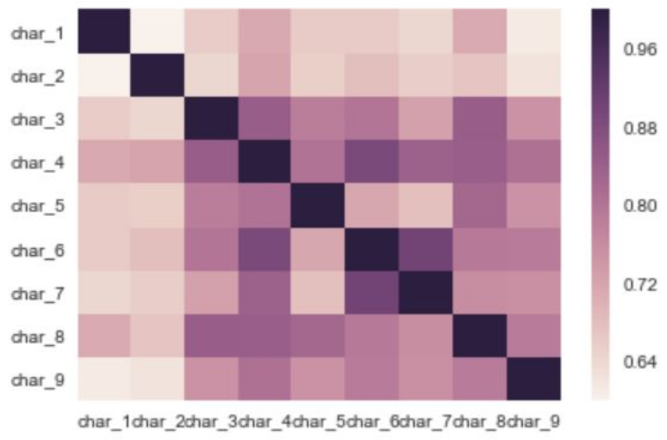
- The people file also contains several different categories of users. Character 1-9, 38 are categorical features of users and Character 10-37 are binary features. Other than those, there is also a group category for each user.

3.2 Data Exploration

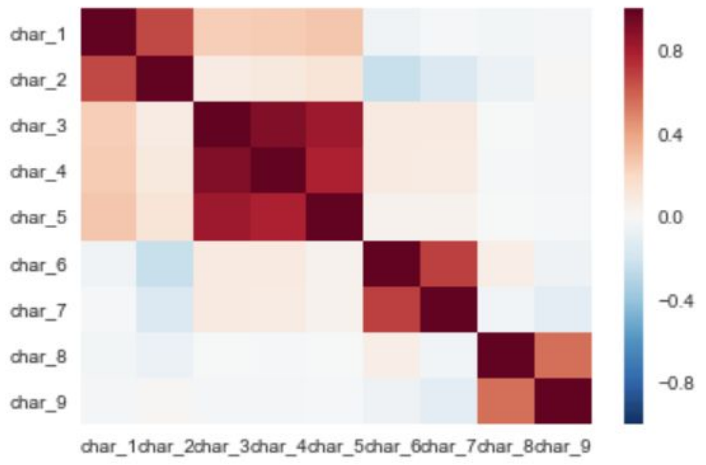
The correlation heat maps were extracted from two feature matrices (activity and user files) separately. From the plot, we could notice that there are high correlations between features from both plots.

In general, we don't want to include all these features (with high correlation) in the model. If there are two or more factors with a high VIF, remove one from the model may be better. Because they supply redundant information, removing one of the correlated factors usually doesn't drastically reduce the R-squared.

But in this project, the model I used is random forest (or gradient boosting framework later), which is resilient to correlated features. In another way, more features may provide more informations. So depending on the goals, sometime multicollinearity isn't always a problem. However, because of the difficulty in choosing the correct model when severe multicollinearity is present, it's always worth exploring.



Correlation Map of Activity matrix

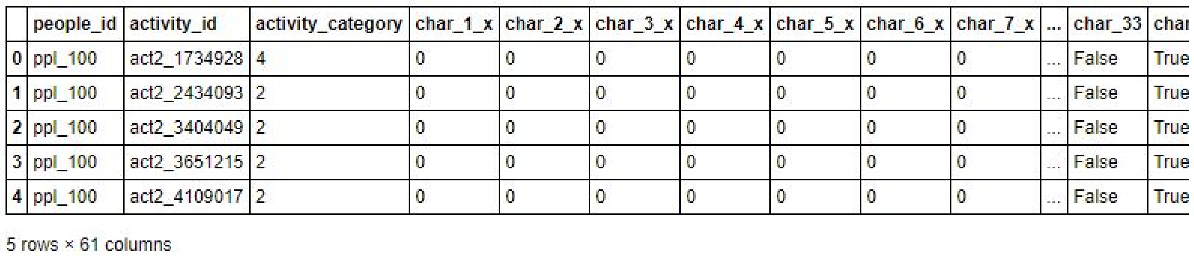


Correlation Map of User matrix

3.3 Data Wrangling

Before fitting the data into the model, I parsed all categorical variables into integer for future processing. Also, I parsed the date into four different features: year, month, day and the weekend(binary).

To develop a predictive model with this data, what I need to do before that is to join the all data together into one single data matrix. The two files can be joined together using person_id as the common key. In this project, I used people file to left join the activity files. Since there are duplicated features, so some features from activity file are marked as 'x' as postfix, and 'y' for features from people file. And after pre-processing of the data, the feature matrix contains 60 features and 1 outcome.



	people_id	activity_id	activity_category	char_1_x	char_2_x	char_3_x	char_4_x	char_5_x	char_6_x	char_7_x	...	char_33	char
0	ppl_100	act2_1734928	4	0	0	0	0	0	0	0	...	False	True
1	ppl_100	act2_2434093	2	0	0	0	0	0	0	0	...	False	True
2	ppl_100	act2_3404049	2	0	0	0	0	0	0	0	...	False	True
3	ppl_100	act2_3651215	2	0	0	0	0	0	0	0	...	False	True
4	ppl_100	act2_4109017	2	0	0	0	0	0	0	0	...	False	True

5 rows × 61 columns

4. Predictive Model Building

In this section, I will try to build a classification model to identifies which customers have the most potential business value, and improve the performance by modifying parameters and selection on features.

4.1 Model Selection

There are several categorical features, which means I will need more features for one-hot encoding, the size of feature map would be extremely big and sparse. Considering such condition, I would prefer using a decision tree classifier. At the very beginning, I considered a random forest model would be fit for it, because there are several advantages of using random forest in this case:

- Random Forest performs better on a data set in high dimension than a linear classifier.
- Random Forest can handle a data set in high dimension without feature selection.
- Also, it can provide features importance ranking after training, which allows me discovering facts and removing some unnecessary features for more accurate results.
- Random Forest needs less time on training than most of other classifiers.

- The generalization error of random forest is unbiased, so it has a better generalization ability to fit the model.
- etc.

4.2 Feature Selection

For training purpose, I would try to use as many features as possible to train the model first. Once I have finished the training process, I can evaluate the result and decide how to parse the feature matrix.

Before training, I have to remove features that are not necessary for training. The feature matrix will be sorted using 'people_id' as key, and the labels will be recorded in an array after that, so that the label and the data sets won't be messed up.

I also separated columns into categorical or noncategorical features for processing and applied one-hot encoding to all categorical features. After that, the feature map became a matrix with over 40,000 columns to be trained.

4.3 Apply a Gradient Boosting model - XGBoost

As founded above, the training data set is extremely large and sparse to be trained. In this case, even using random forest model (sklearn.ensemble.RandomForestClassifier) from sklearn spent lots of time and failed to train at the end.

In order to train this model, I research and found another powerful model, which is XGBoost. XGBoost (eXtreme Gradient Boosting) is one of the most loved machine learning algorithms at Kaggle. It can be used for supervised learning tasks such as Regression, Classification, and Ranking. It is built on the principles of gradient boosting framework and designed to "push the extreme of the computation limits of machines to provide a scalable, portable and accurate library."

XGBoost is one of the implementations of Gradient Boosting model, but what makes XGBoost unique is that it uses "a more regularized model formalization to control overfitting, which gives it better performance," according to the author of the algorithm, Tianqi Chen. Therefore, it helps to reduce overfitting.

5. Observations and Conclusion

There are several observations after running XGBoost model on the data set,

- The model was trained with all available features for the first time running, and the score is 0.978532 and I considered it as the baseline score.
- If I modify the outcome into boolean values, the score drops much. I think that is because the submissions are evaluated on area under the

ROC curve between the predicted and the observed outcome. The penalty is larger if predicted wrong in this case.

- I dropped one of the feature 'char_10' from activity file, which was unnecessary in my opinion after I research the data set. The score was 0.979360, which is a little bit higher than the baseline score.
- But if I dropped another feature 'group_1', the score dropped greatly to 0.903638. Combining the finding above, the first reason could be I dropped too many features from the feature map (from 41,000 to 7000), which makes the model lost too much information. The second reason is that this is also an important feature in the previous trained model. (There are some installation problems, so I am not able to plot the feature importance list.)

After all, since this a competition on Kaggle, so I cannot get the ground truth to evaluate F1-score. The best accuracy score I got is 0.979, which is close to the leaderboard, and this is a great score for me. If further improvement is desired, more time can be spent on adjusting parameters, or parsing the feature matrix, for example, modifying unique terms into the same category, which may help for better model performance.