Machine Learning: SunnyBridge Data Science Case Study

Due on May 31, 2018

Professor Hao Wang

Yifan Chen Ke Zhang Xiaohe He

1 Analysis of the case and dataset

In insurance industrial, customers have the most critical meanings for a company, and their responses directly reflect the effect of market campaign. As the most important targets, customers' responses and related profit are determined by many factors especially their personal background. Data science techniques are exactly used here to estimate and measure future events by the most appropriate model so as to maximize those underlying business efforts for insurance company.
The given training dataset lists the background information of 8137 customers, each of them has 21 kinds of background(features), responses and profit, test dataset lists same features. These features have two basic types: categorical and numerical, some of them are null or unknown. Besides, the sample distribution corresponding with two targets are imbalanced, which also need to be processed. Therefore, our work is to reorganize the given samples and build a prediction model, then use the model to predict responses and customers' profit on testing dataset. Finally, based on both responses result and profit contribution, the market candidates are determined by total profit.

2 Data preprocessing

As one of the most important work, processing the original dataset is required in data science. We need to transform the practical data into scientific form, which can be then used in theoretical experiment. Specificly, in this case, we need to code the categorical features, fill missing values (also named as NA values, i.e, not available), balance samples with different labels (targets) and normalize the numerical features.

2.1 Missing data filling

Missing data filling has different method, which depends on how features perform in the whole dataset. For those features has weak effect on targets, "NA" could be simply deleted, or replaced with mode, mediean or even the prior values. If the features are important towards the targets, regression or other learning methods should be used, that is, to infer them by known data.
There are some features that including "unknown" values, which could be interpreted as: there has existed such values but we don't known (while "NA" means that data is lost), so this type could be added as a new dummy variable within its feature. Here, three features need to be completed and we use two different methods.
1. *custAge*, *schooling*: A reasonable assumption is that a customer's age could definitely affect his decision of whether he would response, and customer's *schooling* could reflect his social status and affect his decision either, so NA in these two features have important information thus need to be obtained by regression. Fig.1 shows the statistical explanation:

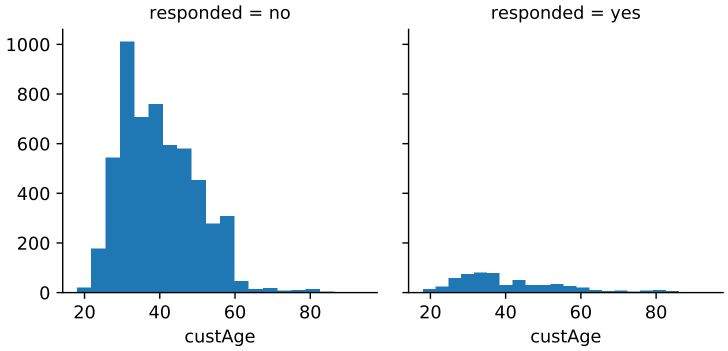

Figure 1: age distribution in responded (both yes and no)

We exact the first four features (*custAge, profession, marital* and *schooling*) to build a subset since they

have strong correlation, then “NA" could be predicted by the regression model. Samples in the subset are divided into 4 groups with different models:
group A: *custAge* with NA and *schooling* without NA;
group B:*custAge* without NA and *schooling* with NA;
group C: *custAge* with NA and *schooling* with NA;
group D: *custAge* without NA and *schooling* without NA.
The only training set we can use is group D with all completed data. Next, ridge regression is applied to predict *custAge* and logistic regression is used to predict *schooling*:
For group A: In group D, we set *custAge* as the single objective and train the set to get ridge regression model, then predict NA age on group A
For group B: In group D, firstly, we split *schooling* as 8 dummy features (or dummy variables, which represents all the subcategories under this feature. We will introduce details in next section), then set these features as objectives. This is a multi-classification problem now (or 8 binary classification), so we fit the logistic regression model towards each dummy features and in each sample, we take maximum probability as its final result. Next, we predict NA schooling on group B.
For group C: In group D, we set *custAge* and *schooling* both as objectives, and train two models to predict missing age and schooling respectively by the same algorithm as group A and group B. Finally we predict NA on group C.
2. *day_ of_ week*: Since it is very likely to be any days in a week, this feature could be filled just using the prior values of each NA.

2.2 Categorical features coding

Dummy variables
Regression treats all independent variables (features) in the analysis as numerical. However, since our dataset include attributes or norminal scale variables such as “profession" or “contract", we can’t distinguish them using only single variables (“0" or “1"), so dummy variables are needed to represent three or more subcategories.
The process of creating dummy variables is equivalent to coding original features as binary format, thus each kind of binary sequence is a subcategory, so all categories could be unique. Note that sometimes, a discretely numerical feature might also be a category due to the feature’s practical meaning.
The categories in our dataset are:

profession, *marital*, *schooling*, *housing*, *loan*, *contact*, *month*, *day_ of_ week*, *poutcomes* and a target *responded*.
After coding them, a group of new dataset is generated with 65 features.
Other processing
1. *campaign*: This feature has two explanation: number of times or number of days. To distinguish them, it is also splited as two numerical subfeatures: *campaign* and *campaign_ days*.
2. *pdays*, *pmonth*: “999" in the two features means client is not contacted, so we set “999" as 0.

2.3 Imbalanced data processing and data normalization

Oversample minorities to balance dataset
Class imbalance occurs in a classification problem when each class does not make up an equal portion in our dataset. It is important to adjust our metrics for the goal, or else, most samples in minority class might be misclassified into majority class and we can still get high accuracy, but such result have meaningless metric in practical context.
In our training dataset, 7310 clients *responded* “no" while only 827 clients *responded* “yes" which is very imbalanced, thus we need to adjust the sample distribution. To make equal portion of two classes, both undersampling and oversampling could be used here, We choose oversampling, i.e., increase minority samples, so as to collect useful information as much as possible.
We use SMOTE algorithm [2]to creating “synthetic" training samples by performing certain operations on real data. The general procedures is for each minority class sample, randomly chosing its k nearest neighbors, introducing synthetic examples along the line segments among the original minority one and its neighbors.

The ratio of line segment can be adjust according to the amount of oversampling required, see Algorithm 1.

Algorithm 1 ALM for SMOTE

```
Input: $T$: training set; $r$: oversampling ratio
for each sample in $T$ do
    find $k$ nearest neighbors as the sample's neighbors
    synthetic new samples between original minority and each of its neighbors according to $r$
    randomly add new samples into $S$
end for
Output: $S$: new training set added synthetic samples
```

Note that after handling data according to previous sections, given training set, Python could automatically adjust the ratio r and output the final new set. The specific code is as following:

```
        from imblearn.over_sampling import SMOTE
        X_resampled, y_resampled = SMOTE().fit_sample(odata, olabel)
```

Normalization

When we finished generating a new training set, the last step before fitting model is normalization, that is, to unify features' metrics, so that the fitting weights distributed to them could be balanced and training accuracy could be guaranteed.

To normalize the training data of numerical features, general methods are introduced as following:

Min-max normalization:

$$\mathbf{x} = \frac{\mathbf{x} - \min}{\max - \min}$$

where min, max are feature's($\mathbf{x}$) minimum and maximum value respectively.

zero-mean normalization:

$$\mathbf{x} = \frac{\mathbf{x} - \mu}{\sigma}$$

where μ, σ are feature's($\mathbf{x}$) mean value and standard value respectively.

Here, we apply *Standardscaler* function in python to preserve features' statistical characters (such as mean value and variance).

3 Models fitting and prediction

Scikit-learn [1] is simple and efficient tool for data mining and data analysis, and we can use it to tackle machine learning problem in Python.

It's a classification problem for us to predict whether the customers respond to the marketing. There are many machine learning algorithms to handle classification problem, such as logistic regression, SVM(Support Vector Machine), RF(RandomForest), DT(Decision Tree), KNN(k-nearest neighbors), naive bayes and GBDT(Gradient Boosted Decision Tree). First, we use 10-fold cross validation to get the accuracy of each algorithm.

Cross validation is used to evaluate the prediction performance of the model, especially the performance of the trained model on the new data, which can reduce overfitting to a certain extent. What's more, we can also get as much useful information as we can from limited data.

Table 1 is the mean accuracy by using 10-fold cross validation.

	Logistic	RF	SVM	DT	KNN	NaiveBayes	GBDT
Accuracy	0.7571	0.9361	0.7921	0.9097	0.8672	0.7363	0.9476

Table 1: the accuracy of different classification algorithm

From the table, we can find that RF, DT and GBDT have high accuracy. Hence here we can use RF and GBDT to predict whether the customers respond to the marketing.

Considering the prediction of profit, which is a regression problem. There are several machine learning algorithm to handle regression problem, such as linear regression, lasso regression, ridge regression and Bayesian regression. In the same way, we use 10-fold cross validation to get the R^2 score, mean squared error and mean absolute error of each algorithm first.
Table 2 is the result by using 10-fold cross validation.

	LinearRegression	LassoRegression	RidgeRegression	BayesRegression
R^2 score	0.9137	0.9197	0.9178	0.9179
MSE(mean square error)	30	29	30	30
MAE(mean absolute error)	1416	1317	1346	1345

Table 2: the evaluation of different regression algorithm

From the table, we can get that they all have a good performance on predicting profit. But if we test linear regression alone by using Hold-Out Method, i.e. all data are randomly divided into two groups, one group accounts for three quarters of the total data as the training set, a group accounts for a quarter of the total data as a validation set. We can judge that it is overfitting since it has a bad performance on validation set. Therefore, we should add regularization items. Here we use LASSO regression to predict the profit in order to avoid overfitting.
By using GBDT and LASSO regression, we can get response and profit of each customers. Then we just choose the customers whose responses are yes and profit is bigger than $30. We can finally get the total profit: 13328. Note that in GBDT, decision trees in each experiment might be different since the subsampling is random, so the total profit might be also different in each experiment. The result we predict and the customers we decide to market are shown in **test_df_GBDT_good.csv**.

4 Feature selection

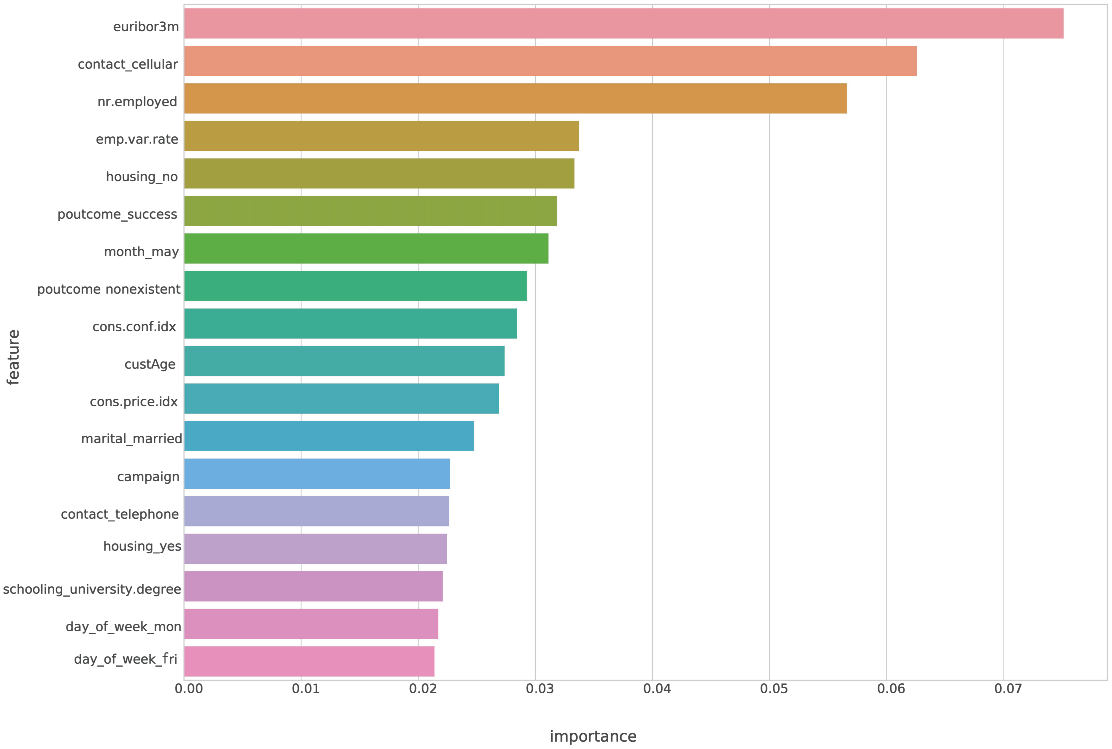

Figure 2: Importance comparision among main features

We have seen relatively high accuracy on the two models using all features, but some of them might have weak weights or has strong correlation with other feature, so such features have little contribution and even could be expressed by their related features. In other words, we could use fewer dominant features to depict the sample space, and still maintain high accuracy. This is called feature selection, which has important value in practical scenario. Fig.2 shows a part of feature of which importance are more than 0.02. Completed importance comparison are shown in appendix. For our client, the insurance company, using fewer features means that they don't need to additionaly collect massive data, which would greatly save their total cost.

5 Model evaluation

Future improvement

- We can use more data to estimate the missing data to attain higher accurate estimation.
- We can try to use less features to predict the responded and profit to reduce complexity. At the same time, it also can guarantee a certain degree of accuracy.

More consideration

We should redefine the profit. As an insurance company, it must consider both the accuracy of predicting responded and the expectation of total profit. For example, even we predict that a customer whose profit is great will not respond to our market, we also want to market him because our prediction is not completely accurate.

$$E(\text{profit}) = \text{profit} \times (1 - \text{accuracy}) + (-30) \times \text{accuracy} \quad \text{if responded} = 0$$

$$E(\text{profit}) = \text{profit} \times \text{accuracy} + (-30) \times (1 - \text{accuracy}) \quad \text{if responded} = 1$$

First, we should set a threshold for accuracy, because low accuracy means high risk. This depends on how much risk the insurance company is willing to assume. Then we think of the real profit we get. If $E(\text{profit}) >= 0$, we will decide to market the customer.

$$\text{real profit} = \sum_{\text{marketed}} E(\text{profit})$$

Finally we must choose the scheme that has maximal real profit.

Reference

[1] User guide: contents — scikit-learn 0.19.1 documentation. `http://scikit-learn.org/stable/user_guide.html`. (Accessed on 05/31/2018).

[2] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

Appendix
Importance comparision

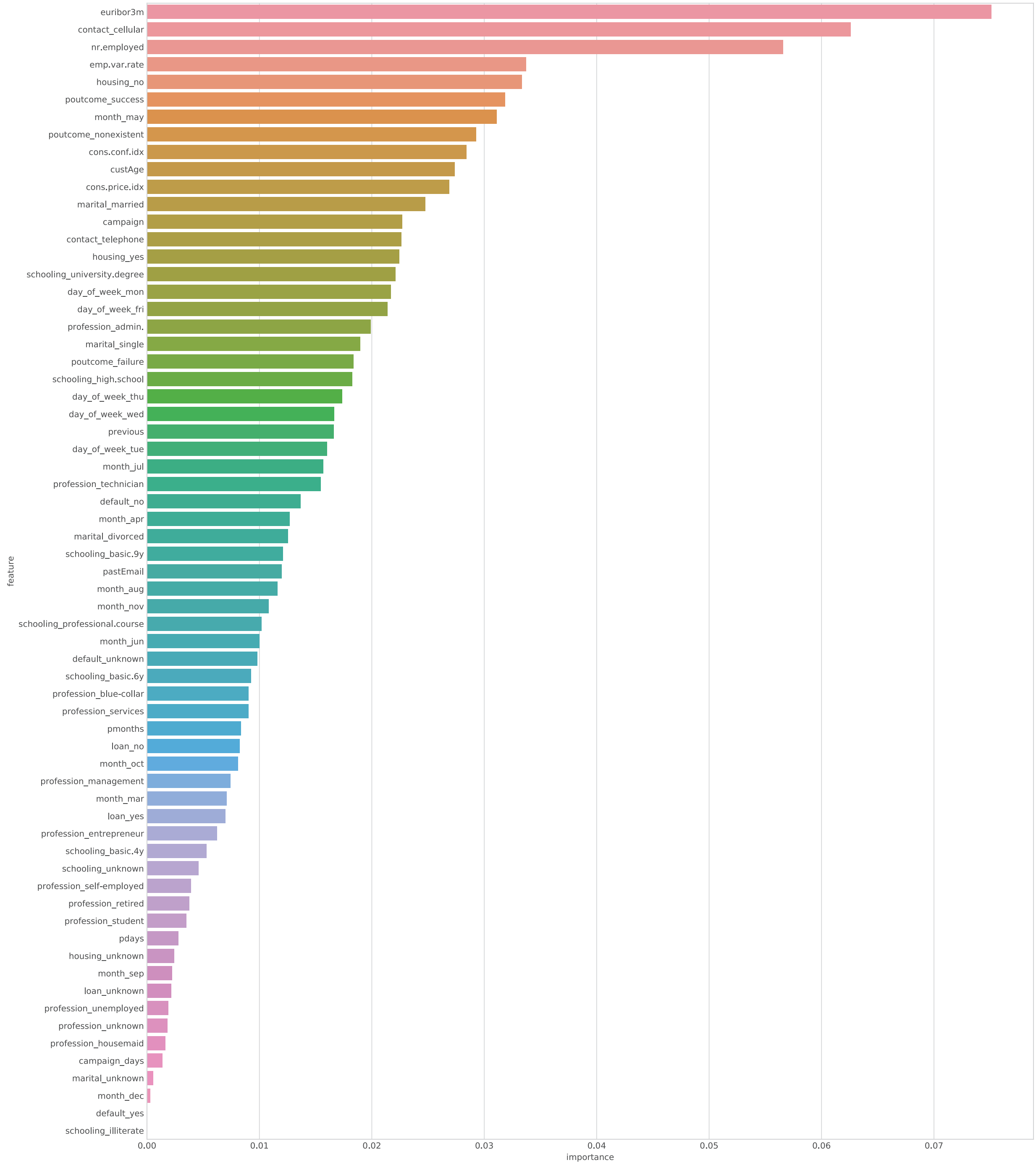

Figure 3: Importance comparision among all 65 features