Online Consumer Behavior Analysis and Prediction Through Machine Learning

BOFAN DONG 450418239

CHAOMIN YUAN 460126294

MANQI QU 450039333

MENGDIE XU 450594577

ZHONGSHU LIN 450628986

Abstract

This report intends to predict online consumer behaviors through machine learning algorithm in particular classifier training and testing. Through comparison with benchmark KNN classifier, the work shows that Quadratic Discriminant Analysis under Bayesian Law and Logistic classifier both outperform and can provide statistically and economically significant results. The learning process in this report is expected to be repeated on any suitable dataset and provide meaningful predictions for online business operation.

Keyword: machine learning, classification, Bayesian classifier, Logistic Classifier

1, Background

With the booming of the Internet era, ecommerce has become a major sector and changed the traditional retail industry dramatically with low sales cost and intense marketing events. Massive promotions such as “Single’s Day” created by Chinese ecommerce giant Alibaba have become a widely accepted and practiced competition point. One problem aroused from this situation is the effectiveness and efficiency of these marketing events, which constitute a large proportion of online retailer’s costs. A successful marketing event would expect to convert new customers into repeated buyer eventually in his or her life cycle instead of one-time discount hunter. Therefore prediction on future consumer behavior becomes essential in order to evaluate and improve these events. Fortunately, one of the new features of ecommerce is the rich data pool on valuable information such as demographic profile, interactive behavior, transaction history, etc., which can be used to learn and predict whether a customer will be converted into repeated buyer for certain merchant through certain event. With a well-trained model, online retailers will be able to target more precisely on high conversion probability customers therefore improve the marketing efficiency and business profitability in general.

Thus, the aim of this report is to generate and evaluate some of the possible models based on the given data and to create a repeatable learning process for future use of any other stakeholders. Probabilistic generative process will be applied on customer features include demography, interaction, and purchase history to give a symbolic probability of conversion and then selected decision rule will classify this given customer into one of the two categories: conversion and non-conversion, which are represented by the predicting target label 1 and 0.

2, Data Description and Features Engineering

Data comes from the Chinese ecommerce company Alibaba (website reference) and consists of 6-month interactive activities records of each User ID with the website up to a big promotion event date, which can be break down into time stamp, action type, involved brand, involved item and item category. Demographic features including age category and gender are attached to each User ID. Although this data is sampled in a biased way in order to keep business confidentiality, the applicability of the modeling process will not be influenced.

The complete dataset provided includes 7 million records under 5000 merchant IDs. Considering that customer behavior may vary for different merchants as well as to control dimensionality, records under one single merchant with merchant ID 3828 are selected. We ignore involved brand and item information because it is not relevant to the purpose of this report. Another issue with the data is that multiple activities are executed by same User ID therefore multiple categories can be assigned to this same User ID if these activities are used as different data entries. Although this is implementable by aggregating either the predicted probability or the predicted category labels for each User ID in the end, we decided to move the aggregation ahead

onto the dataset grouping by User ID. Action counts and Timestamp are summed up for identical User ID and new features are created as shown below. It is also economically and logically sensible given that periodical behavior pattern usually provide more information than an isolated action. For example, the conversion of a certain user is more likely to be explained by a high activeness on the website and a reasonably large amount of actions rather than a single purchase.

Number_of_0: Total 'clicks' on the website
Number_of_1: Total 'add-to-cart' on the website
Number_of_2: Total 'purchase' on the website
Number_of_3: Total 'add-to-favorite' on the website
Total: Total actions of all four types on the website
In_number_of_0: Total 'click' under merchant ID 3828
In_number_of_1: Total 'add-to-cart' under merchant ID 3828
In_number_of_2: Total 'purchase' under merchant ID 3828
In_number_of_3: Total 'add-to-favorite' under merchant ID 3828
In total: Total actions of all four types under merchant ID 3828
Total_active_days: Total days on which any action is executed on the website
Recent_active_days: Total days in November i.e. event month on which any action is executed on the website

3, Feature Selection and Cross-Validation

Feature selection: there are several reasons to perform a feature selection. First of all, the twelve new features generated above can be correlated to one another; for example, the amount of total action is a linear combination of the amount of different types of action. As correlation usually introduces biases and problems to modeling process, eliminating these redundant features is necessary. Secondly, dealing with high-dimensional data is computationally intensive and can take a long time to apply the algorithm. It is also a trade off between the variance and bias of models in order to avoid over fitting problem and to improve model interpretability. Lastly, identifying most significant features is also aligned with business logic that online business owners would prefer to spend the limited resources monitoring and analyzing the most influential factors so as to improve operation efficiency.

In general, there are two ways of selecting features – filter method and wrapper method. The former is independent from any machine learning algorithm, while the latter combines the

classification along with the feature selection process and the result of this classification will be the evaluation of the feature selection with wrapper method. This report will use the filter method given the computational ability limitations

Data is grouped by the 'label' column and all other numerical features are then aggregated by within group average. A ranking score is introduced with the following formula for each feature:

$$S = abs(log_2(f_{1i}) - \log_2(f_{0i})), where\ i = 1,2 \dots 12$$

Here S is the ranking score where 'abs' is the absolute value function and f_{1i} is the average of the ith feature within group label 1 and same applies for f_{0i}. The six features (as shown below) with the highest ranking scores S are then selected as the final model inputs.

In_number_of_0
In_number_of_1
In_number_of_3
total_active_days
number_of_0
in_number_of_2

Cross validation:The essential part of model fitting is to estimate the parameters of each model. However, these parameters are usually estimated by either minimizing errors or maximizing likelihood in training set and therefore, it is not uncommon to see models perform well in training samplewhilegeneratepoor results on the testing data. In the meanwhile, when the training process attempts to minimize in-sample error, over fitting problems can easily arise. As traditional validation method tends to lose significant information while slicing a chunk of available data into testing set, cross-validation will be introduced and applied to our dataset in order to test the stability of model performance especially out-of-sample and to prevent potential over fitting issue.

The most widely practiced cross-validation methods include exhaustive methods such as leave-p-out cross-validation, leave-one-out cross-validation and non-exhaustive methods such as k-fold cross-validation. Exhaustive methods incorporate the most available information in model training but are usually time consuming. K-folder provides a reasonable balance between data incorporation and computational complexity.

Therefore in this report, a six-folder validation will be employed and the dataset is randomly partitioned into six equal size folders. Each folder then will take turns to be testing set while the rest five folders will altogether be training set. With this validation, every data entry contributes to both model training and model validating. Lastly, the six estimated results will be simply averaged to generate the final estimation.

4, Model Selection

k-Nearest Neighbor (k-NN):k-NN classification was developed from the demand of performing discriminant analysis when reliable parametric estimates of probability densities are unknown. In an unpublished US Air Force School of Aviation Medicine report in 1951, Fix and Hodges (1951) introduced a non-parametric method for pattern classification that has since become known the k-NN rule. They introduced a novel approach to nonparametric classification by relying on the 'distance' between points or distributions. The basic idea is to classify an individual to the population whose sample contains the majority of nearest neighbors. Later in 1967, some of the formal properties of the k-NN rule were worked out.

The k-NN is the fundamental and simplest classification technique when there is little or no prior knowledge about the distribution of the data. Given a new data entry and an observed training set, all the distances between the new data and all data points in the training set can be computed. Ak-NN classifier ranks these distances and aggregates the categories of k nearest neighbors as the prediction result for the new data point.

Performance of a k-NN classifier is primarily determined by the choice of k as well as the distance metric applied. A little thought suggests that the error on the training data should be approximately an increasing function of k, and will always be 0 for k = 1, which is known as the Nearest Neighbor rule (NN). Sum-of-squared errors generally does not work as a criterion for picking k because it would always pick k = 1 (Hastie, Tibshirani, & Friedman, 2009), which is usually an over-fitted estimation. Nevertheless, a large value of k makes the estimate over smoothing and the classification performance degrades easily with the introduction of the outliers from other classes (Imandoust and Bolandraftar, 2013).

Although the k-NN algorithm is among the simplest of all machine learning algorithms (Haara and Kangas, 2012) and makes no distribution assumptions on original data, it is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The choice of k i.e. the tradeoff between model variance and model bias is also difficult. Therefore it is usually used as a starting point and referencing benchmark for more advanced classifiers, as it is in this report. This model can also be justified considering that customers with similar demographic characteristics and past behavior patterns are more likely to have similar future actions.

Logistic Regression:Logistic regression (also known as logit regression) developed by statistician David Cox in 1958, is a regression model which estimates probabilities to measure the relationship between categorical response variable and one or more explanatory variables by using logistic function (Hosmer and Lemeshow 2005). The logic behind this is to find the parameters β to best fit y =1 when $\beta_1 x_1 + \beta_2 x_2 + \ldots + \varepsilon > 0$ or y=0 when else (Hosmer and Lemeshow 2005), where β is unknown parameter and ε is error. By conducting the estimate

approach called ‘maximum likelihood’, logistic regression can model the posterior probabilities of the K classes via linear function in x, and ensure the sum of it remain in [0,1] (Hastie, Tibshirani, & Friedman 2008) Similar to other regression model, logistic regression can use one or more continuous or categorical variables to predict binary dependent variables. To convert binary variables to continuous variables, we use odds ratio of event happening for different levels of each independent variable, then take logarithm of that ratio to create continuous criterion. Although logistic regression model can be binomial (binary), ordinal or multinomial, considering the required response is only classified as two classes: repeating buyer or no repeating buyer in this project, we simply introduce binary logistic regression model, the two-class case (K=2), in detail. When K=2, logistic function is defined as $F(x_1) = \frac{1}{1+e^{-(x^T\theta)}}$ and $F(x_2) = \frac{e^{-(x^T\theta)}}{1+e^{-(x^T\theta)}}$ where θ is parameter.

The underlying assumptions in this model is that the outcome must be discrete, which means the dependent variables should be dichotomous in nature; there should be no outliners in the data; there should be no high correlations among predictors, which means variables should be independent from each other. Also, there should be linear relationship between the odds ratio and each independent variable. In this report, our data generally fits the underlying assumptions of logistic regression: our case has two discrete outcomes – conversion and non-conversion; outliners are excluded in cleaning data process; independence between features however is not well satisfied since different types of actions can and usually have influence on each other.

The reasons why we choose logistic regression model are plenty. Logistic regression model is popularly used in analysing business market behaviour as its many characteristics. From foundation logistic function aspect, there are two main reasons contribute to its popularity. First is that the estimates always lie within the range of one to zero (Kleinbaum and Klein 2010). This is aligned with characteristics of probability and is easy to understand and interpret. Second is that the S-shaped description of model combines effects of several factors on the projected topic. S-shaped of f(z) indicates the probability on the result is minimal at beginning until some threshold is reached, then the probability will have increasingly grow while z is in the range of intermediate and remain extremely high around 1 once z gets sufficiently large. z represents index of combined influenced factors. This also can be interpreted in practise as customer behaviour results are altered by combined variables and set of threshold (Kleinbaum and Klein 2010).

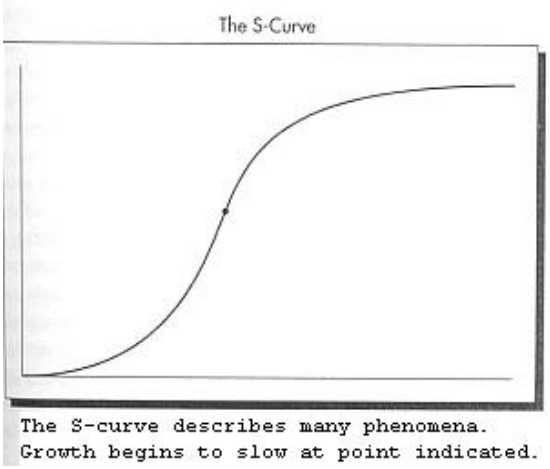


figure 1

Most importantly, the assumptions under logistic regression are very robust, which means researchers do not have to examine whether all their data fit linearity, normality and equal variance assumptions in each group. It is also not assumed that the error term variance is normal distributed (Tanguma and Saldivar, 2010). The ease of assumptions may increase the convenience of logistic regression being applied in real life. Hence, interpreting estimated coefficients as adjusted log odds ratio is easy to understand and use practically.

There are also several limitations related to logistic regression model. Depending on its characteristics, it cannot predict continuous outcomes for researchers. Also, underlying assumptions can be flawed sometimes. Overfitting is another issue with which the model appears to have sampling bias and becomes overconfidence in prediction.

Bayesian Classifier andQuadratic Discriminant Analysis: QDA is a common approach to solve supervised classification problems and specifically a general form of Bayes Discrimination. This model uses the estimation of the likelihood of each class as a Gaussian distribution, afterwards uses the posterior distributions to estimate the parameters in the tested data (Hastie, Tibshirani and Friedman 2013). For example, Maximum Likelihood Principle can be used to estimate the Gaussian parameters with the likelihood derived from training data (Srivastava, Gupta and Frigyik 2007). As one of the special examples of Bayes Theorem, Quadratic Discriminant Analysis is used to determine which variables discriminate the set of data, which is supposed to be classified.

Under the assumption of $X|G = k \sim \mathcal{N}(\mu_k, \Sigma_k)$, the density is

$f_k(\mathbf{x}) = \frac{1}{(2\pi)^{p/2}|\mu_k|^{1/2}} \exp\{-\frac{1}{2}(\mathbf{x} - \mu_k)^T \Sigma_k^{-1}(\mathbf{x} - \mu_k)\}$, then a unit with feature vector **x** is assigned to the class which maximizes the logarithm:

$$\log\left(\pi_k \mathrm{f_k}(\mathbf{x})\right) = \log \pi_\mathrm{k} - \frac{1}{2}\log|\Sigma_\mathrm{k}| - \frac{1}{2} d(x; \mu_\mathrm{k}, \Sigma_\mathrm{k})$$

In terms of the last component of this logarithm, $d\left(x; \mu_\mathrm{k}, \Sigma_\mathrm{k}\right)$ equals $\left(x - \mu_\mathrm{k}\right)^T \Sigma_\mathrm{k}^{-1}(x - \mu_\mathrm{k})$, which is the Mahalanobis distance of each observation from the centroid of the k-th group (Mahal 2016).

Going back to the logarithm, we name $\log\left(\pi_\mathrm{k} \mathrm{f_k}(\mathbf{x})\right)$ as $\delta_\mathrm{k}(\mathbf{x})$, which refers to a quadratic discriminant function. In order to find the $\delta_\mathrm{k}(\mathbf{x})$, which maximizes the classifier $\widehat{\mathrm{G}}(\mathrm{X}) = \mathrm{argmax_k}\{\delta_\mathrm{k}(\mathbf{x})\}$, it is essential to specify the prior probabilities, π_k, and the within group covariance matrix, $\widehat{\Sigma}_k$. However, if π_k is not related to *k*, **x** is assigned to the group to which the Mahalanobis distance is least.

Linear discriminant analysis is also a widely used form of Bayesian classification and it is distinguished from QDA through the trade-off on model bias and model variance. Assuming there is *p* predictors, quadratic discriminant analysis requires a covariance matrix for each class and in total$Kp(p + 1)/2$ parameter estimations. As for linear discriminant analysis, it assumes that *K* classes share a common covariance matrix hence only requires estimation of *Kp* linear coefficients. Moreover, quadratic discriminant analysis can be simplified as linear discriminant analysis when $\Sigma_k = \Sigma$. As a result, QDA tends to have larger flexibility and a higher model fitness than LDA in the meanwhile it also induces large model variance. It is more appropriate when there is a large training data set (James, Witten, Hastie, and Tibshirani 2015), as is for our case.

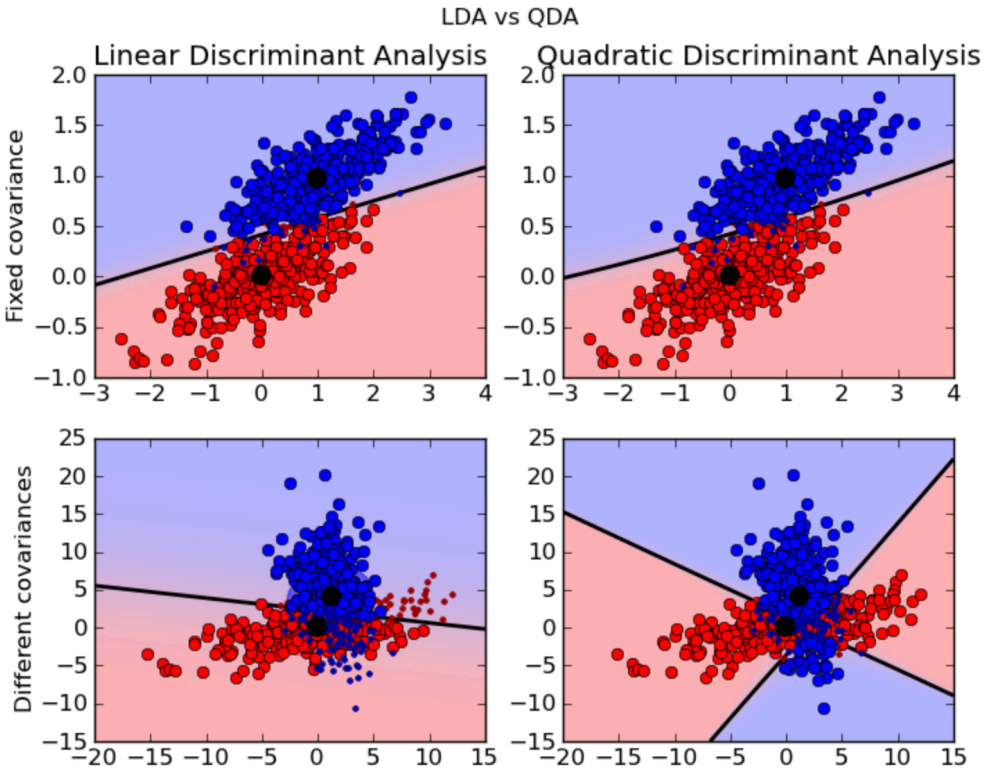


figure 2

Quadratic discriminant analysis is a greatly outstanding tool to solve the classification problems. Considering the training data set we have chosen and the feature characteristics, it is unlikely for all eight features to share the same covariance matrix. As illustrated above, when the occurring groups have different covariance matrices, quadratic discriminant analysis is more favorable and suitable.

Evaluation Criteria: one issue with our dataset that it is highly imbalanced. The ratio of positive (conversion) samples verse negatives (non-conversion) is around 1 to 9. A problem resulted from such imbalanced dataset is that the normal measurement such as accuracy score, F1 score or specificity and sensitivity may not describe the performance of the models as well. For example, if a classifier predicts all the test samples as negative, it will reach a high accuracy score at around 90 percent. But apparently there is no valuable information. Another problem is that in some model which base on a score to discriminate test samples, the different threshold setup will lead to different results, which makes it difficult to measure the performance of such models. For example, in the classic load default classification problem with a logistic classifier, the threshold of load default can be set as 0.5 in normal case or can be set as 0.1 if the bank tends to make a conservative estimation. However, the different threshold will lead to a different accuracy score as well as sensitivity and specificity.

This report adapts a roc curve and auc score as the performance criteria for model evaluation, which address to and solve problems discussed above.

In roc curve diagram, true positive rate (i.e. sensitivity) is plotted against false positive rate (i.e. 1- specificity) for each model with regards to all possible cutoff values. Usually the threshold will be set from 0 to 1, and the step can be set as 0.1 or 0.01, the smaller the step, the more accurate of the roc curve will be. The optimal threshold i.e. cutoff value is corresponded to the point at the upper-left corner, which represents sensitivity and specificity are both 100% that the model gives perfect prediction. This (0,1) point is called perfect point. To compare performance of different models, the roc curve that locates on the most northwest or that has the largest under-curve-area is corresponded to the best model. The AUC score measures the area under the ROC convex hull. 0.5 is a widely accepted threshold to define whether a model is well performed or badly performed.

With the incorporation of roc curve and auc score, we are able to:

1) Compare the general model performance on the scale of all threshold choices i.e. a complete picture instead of taking a snapshot at a specific threshold value and determine model performance.
2) Eliminate the problems caused by imbalanced dataset without manipulating original data. Regular methods of dealing with imbalanced dataset such as under-sampling and over-sampling edit directly on the original data and therefore may change the underlying information in the meanwhile. However, imbalance is no longer a problem if the model has adjusted itself to reflect this imbalance information, i.e. chooses threshold value accordingly instead of the general over-or-below-0.5 rule. In the sense that roc curve presents a complete picture of model and therefore any possible adjustments are already included.

5, Model Estimation and Results Analysis

When models are estimated on training data through algorithm, predictions are then generated on testing set in ways that: 1) for kNN, the model output is the average of labels of the k nearest neighbors and will be assigned ‘1’ if the output is greater than 0.5 and ‘0’ otherwise; 2) for logistic regression and Bayesian QDA, the model output is the probability of being label ‘1’ and will be assigned ‘1’ if the outputs is greater than a threshold probability. Results of the first cross validation folder are presented and being analyzed as an example.

kNN: we apply kNN to calculate data in group1 and the k are chosen from 1 to 20. As showing following, the accuracy rate of k is almost stable when k>6, thus, we could choose k=10.

```
In [39]: knn_model_selection(0, 20)
Out[39]:
[0.7716390423572744,
 0.8545119705340702,
 0.8250460405156538,
 0.8674033149171271,
 0.850828729281768,
 0.8784530386740331,
 0.8710865561694291,
 0.8784530386740331,
 0.8766114180478821,
 0.8784530386740331,
 0.8747697974217311,
 0.8784530386740331,
 0.8766114180478821,
 0.8784530386740331,
 0.8784530386740331,
 0.8784530386740331,
 0.8747697974217311,
 0.8784530386740331,
 0.8784530386740331,
 0.8784530386740331]
```

figure 3

Confusion matrix: confusion matrix is a popular way in measuring model predictability and to incorporate personal preference based on the prediction purpose. Three tables below are confusion matrices when threshold probability for logistic and QDA are set equal to 0.5, 0.9, and 0.1 respectively. 0.5 is the regular threshold widely used in practice and it is shown that logistic classifier gives the highest accuracy. However, considering the specific topic background, the business would rather bear higher type 1 error (i.e. false positive) than suffer from large type 2 error (i.e. false negative). In other words, reaching out and communicating with more potential conversion (predicted positive) is usually more favorable than missing out high potential customers. Therefore, minimizing false negative rate becomes more desired than increasing overall accuracy. With threshold of 0.5 QDA has the lowest false negative rate.

	Prediction	KNN		Logistic		QDA	
		Positive	Negative	Positive	Negative	Positive	Negative
Actual	Positive	0	66	1	65	11	55
	Negative	3	474	3	474	33	444
	Accuracy	0.8729		0.8748		0.8379	

In the meanwhile, a threshold probability of 0.9 is applied in proportion of the distribution of labels in our original dataset. This application is expected to improve the overall accuracy, which is confirmed by the confusion matrix below. However, this improvement mainly comes from

lower false positive and higher true negative, which is less cared regarding to our topic. False negative rates both increased. QDA still gives the least false negative predictions.

	Prediction	KNN		Logistic		QDA	
		Positive	Negative	Positive	Negative	Positive	Negative
Actual	Positive	0	66	0	66	4	**62**
	Negative	3	474	1	476	24	453
	Accuracy	0.8729		**0.8766**		0.8416	

Now incorporating preference of the specific topic i.e. desire for low false negative rates, a low threshold probability of 0.1 is applied and results are presented below. Although overall accuracy dropped largely, false negative rates for both models also plunged especially for logistic classifier (from 12% to 1%). In this situation logistic classifier beats QDA and gives the least false negative predictions.

	Prediction	KNN		Logistic		QDA	
		Positive	Negative	Positive	Negative	Positive	Negative
Actual	Positive	0	66	59	**7**	19	47
	Negative	3	474	392	85	78	399
	Accuracy	**0.8729**		0.2652		0.7698	

In general, prediction accuracy over 80% represents a well-performed model and logistic regression model among all has the best performance. QDA underperforms benchmark kNN probably due to the underlying likelihood distribution assumption, possible improvement of which is further discussed in limitation and improvement section.

ROC curve and AUC score: in order to integrate what has been discussed and to generate a landscape view of model performance on different threshold values, roc curves are drew and auc scores are calculated. Results of experiments on six CV folders are presented below.

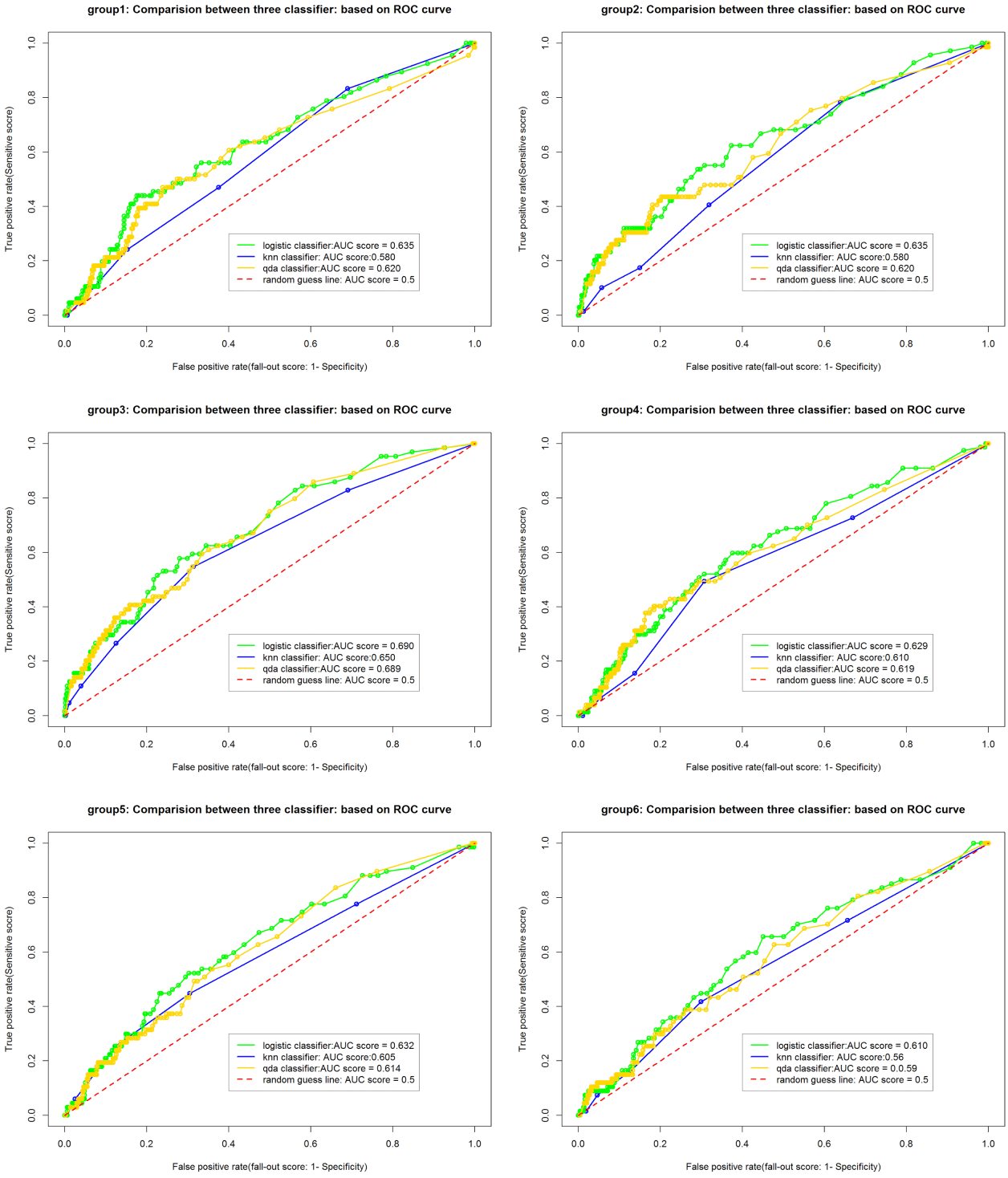


figure 4

As is shown, both Logistic and Bayesian QDA classifier have similar performance: the average AUC score of both model will be around 0.65, While the average AUC score of Logistic classifier is 0.6513 and the average AUC score of QDA is 0.6511. As for the KNN model, it has a relatively weak performance and its average AUC score is around 0.6. Because that KNN just

simply calculate the ratio of data points with the k nearest neighbors, it will always return several repeated probabilities. In our project, the probabilities that KNN return will be 1/10, 2/10, etc. With only several probabilities to predict, we can get limited pairs of true positive rate and false positive rate combination. Those fewer pairs of data may not be sufficient to plot an accurate roc curve, which may underestimate the performance of KNN.

6 ,Conclusions

Through the complete learning process and results analysis, this report has reached several conclusions:

1) When predicting future behaviour of customer under a given merchant, the interactive record between this customer and the merchant turned out to be more relevant than the demographic information of the customer.
2) Original data suggests that female customers consist of a majority of online customer base, which should provide the business some insights in events design and customer appealing.
3) It is statistically significant and economically meaningful to model and predict customer behaviour with a rich data library and robust algorithm, as is proved by our model performance analysis. Machine learning can and should play a major role in ecommerce industry.

7, Limitations and Further Improvements

Although the selected models in general have good performance, there are some limitations. First of all, all numerical features in original dataset show long tail and positive skewness (active days as an example presented in figure below and others demonstrate similar pattern). Also sample distribution in terms of different gender is heavily biased on female (value 0) with over 80% percentage. Although this is in consistence with the economic fact that online shopping is more popular among female, it can still cause problem for our models given that much less information is available on the male group. Further feature engineering can be explored and applied to original data, which should improve the learning process and prediction results.

Secondly, there are plenty of improved forms of the selected models that can be applied with necessary computational skills. For example in logistic regression, a kernel function can be used instead of the simple linear component, which as a non-parametric and data dependent method will interpret the information contained in original dataset to the largest level. Similarly, instead of assuming normal distribution for the likelihood of each class, the distribution can be estimated through a kernel approach and therefore achieve much closer probability density functions to the real ones.

With the deepening understanding of data and the developing of relevant algorithms, these improvements will be easily implemented someday and hopefully provide more contributions to improve our business and everyday life.

Reference List

Fix, E. & Hodges, J.L. 1951 "Nonparametric Discrimination: Consistency Properties",*RandolphField, Texas*, Project 21-49-004, Report No. 4.

Haara, A.; Kangas, A. 2012 Comparing k Nearest Neighbors Methods and Linear Regression—Is There Reason to Select One over the Other? *Math. Comput. For. Nat. Res. Sci.* **2012**, *16*, 50–65.

Hastie, T., Tibshirani, R. and Friedman, J. 2013, *The Elements of Statistical Learning:Data Mining, Inference, and Prediction*, Springer Science & Business Medi, Berlin.

Hosmer, D. W. and Lemeshow, S. 2000, *Introduction to the Logistic Regression Model,* in Applied Logistic Regression, Second Edition, John Wiley & Sons, Inc., Hoboken, NJ, USA.

James, G., Witten, D., Hastie, T and Tibshirani, R. 2015, *An Introduction to Statistical Learning with Applications in R*, Springer Science & Business Medi, Berlin.

Kleinbaum, D.G. and Klein, M. 2010, *Logistic Regression a self-learning text*, Third edition, Springer, Atlanta GA.

Mahal 2016, MathWorks, Natick, viewed 13 October 2016.

<https://au.mathworks.com/help/stats/mahal.html#responsive_offcanvas >

S B Imandoust et al. Int. 2013, *Journal of Engineering Research and Applications,* Vol. 3, Issue 5, Sep-Oct 2013, pp.605-610

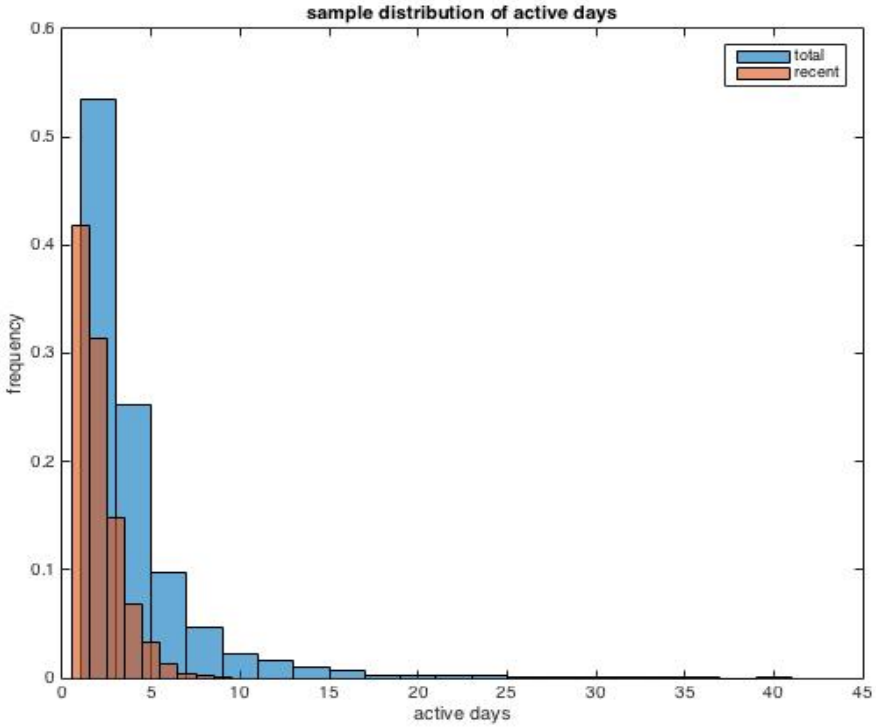
Srivastava, S., Gupta, M. and Frigyik, B. 2007, 'Bayesian Quadratic Discriminant Analysis', *Journal of Machine Learning Research*.

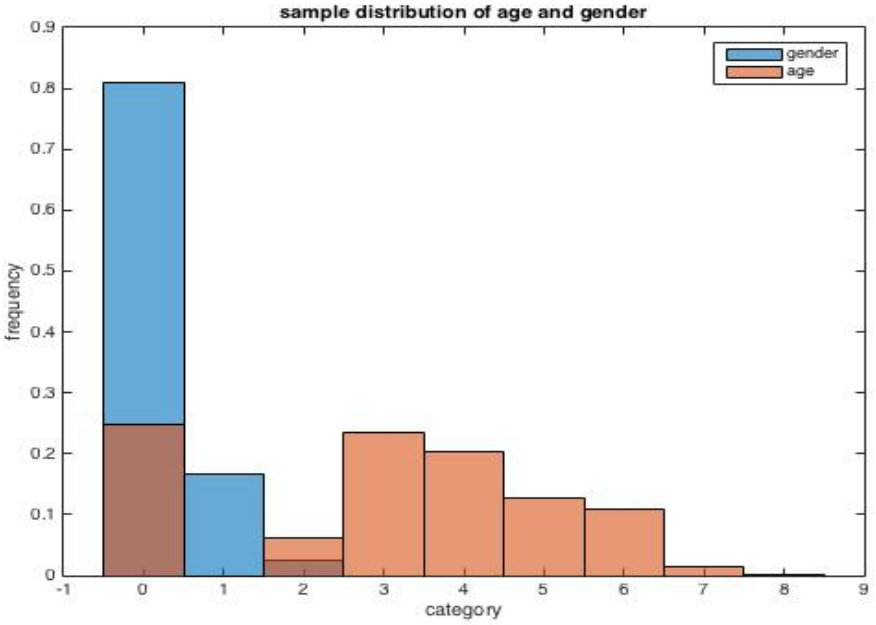
Tanguma, J. and Saldivar, R. 2010, *INTERPRETATION OF LOGISTIC REGRESSION MODELS IN MARKETING JOURNALS*, The university of Texas-Pan, US.

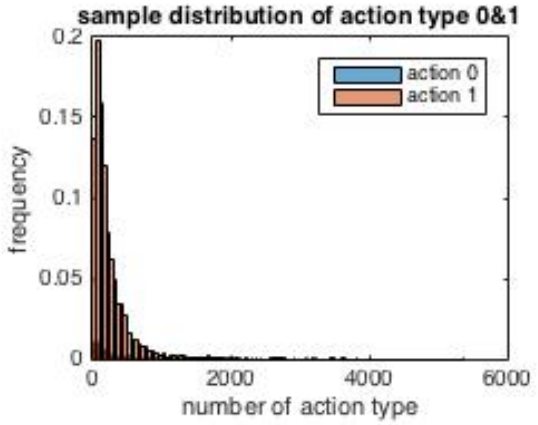
T. Hastie, R. Tibshirani and J. Friedman. 2009, 'The Elements of Statistical Learning: Data Mining, Inference and Prediction', *Springer Series in Statistics*, 2nd Ed. Springer.

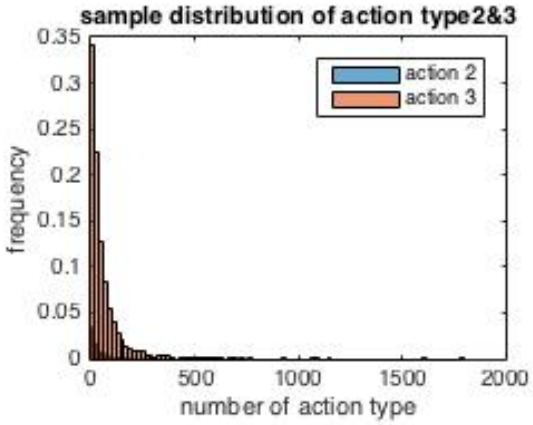
Appendix

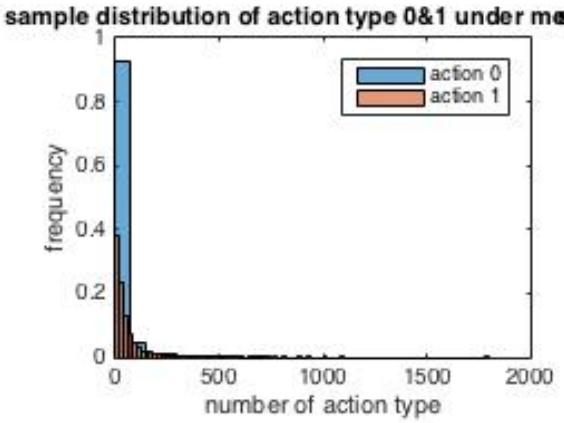
(1) data description graph

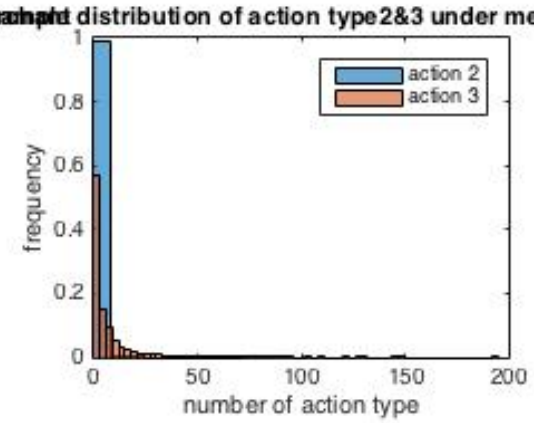


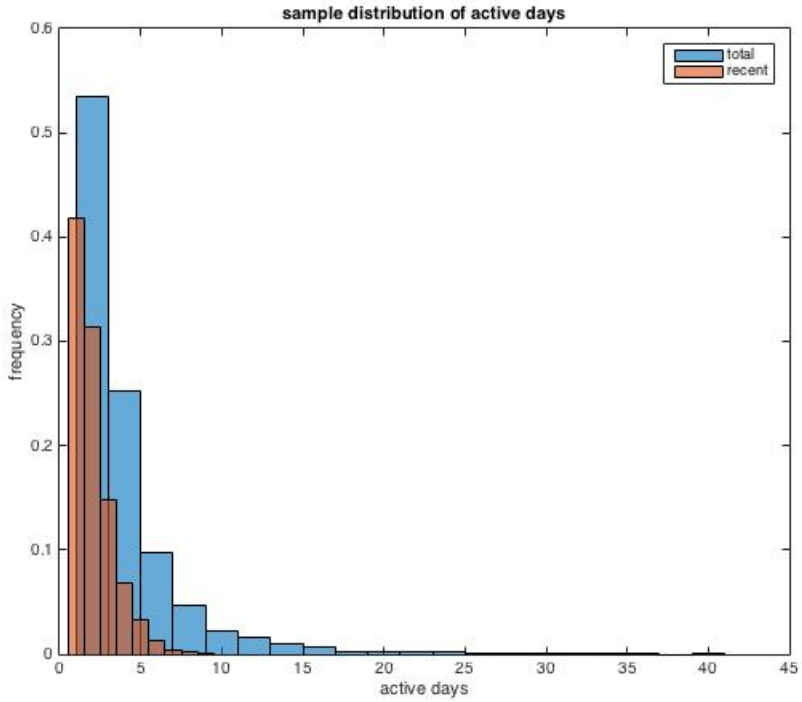


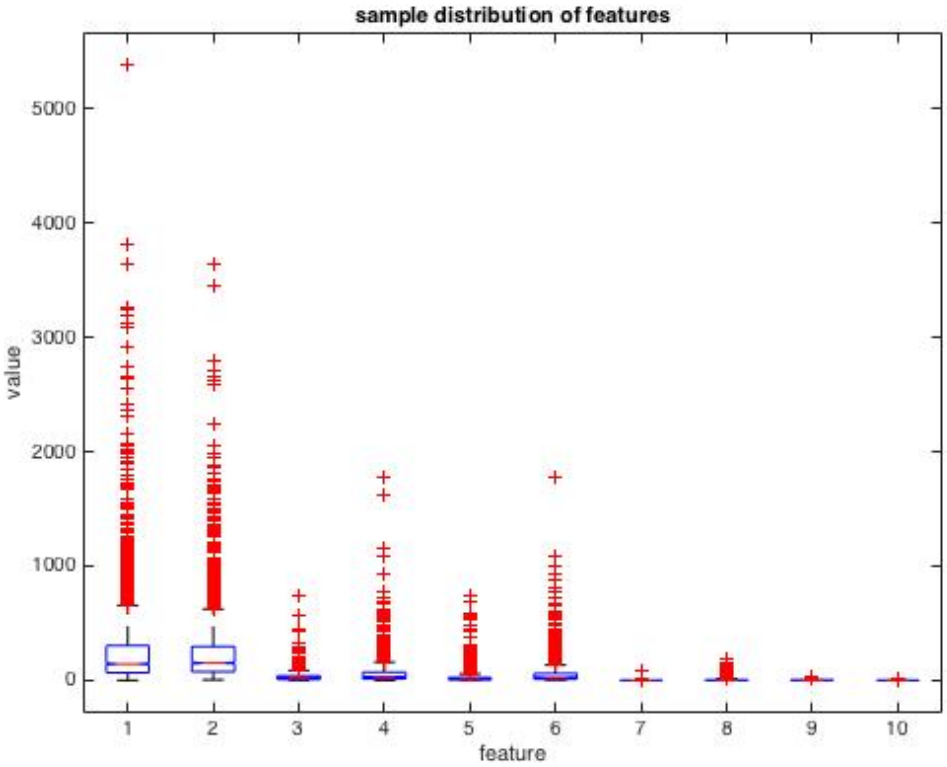

sample distribution of action type 0&1
action 0
action 1
frequency
number of action type
0
0.05
0.1
0.15
0.2
0
2000
4000
6000

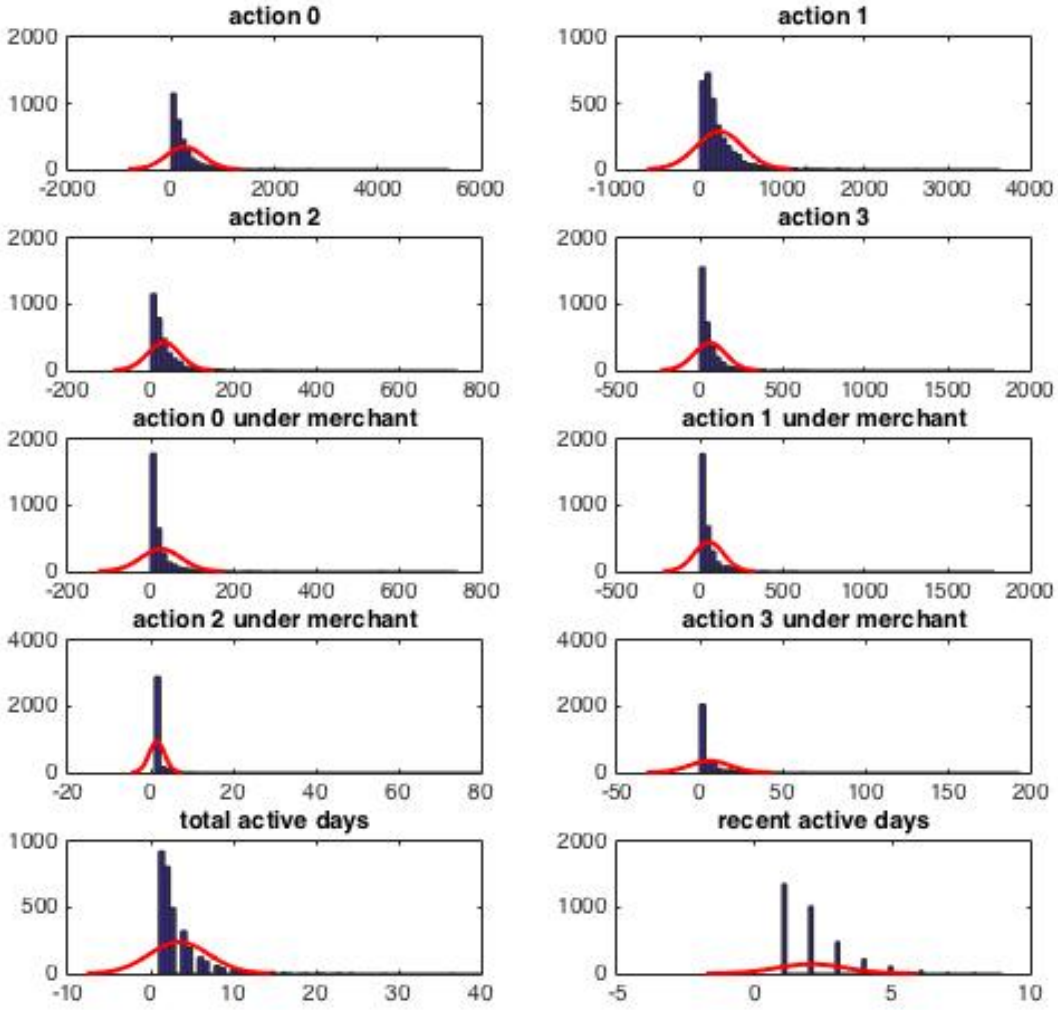

sample distribution of action type2&3
action 2
action 3
frequency
number of action type
0
0.05
0.1
0.15
0.2
0.25
0.3
0.35
0
500
1000
1500
2000


sample distribution of action type 0&1 under merchant
action 0
action 1
frequency
number of action type
0
0.2
0.4
0.6
0.8
1
0
500
1000
1500
2000


distribution of action type2&3 under merchant
action 2
action 3
frequency
number of action type
0
0.2
0.4
0.6
0.8
1
0
50
100
150
200


sample distribution of active days
total
recent
frequency
active days
0
0.1
0.2
0.3
0.4
0.5
0.6
0
5
10
15
20
25
30
35
40
45


sample distribution of features
value
feature
0
1000
2000
3000
4000
5000
1
2
3
4
5
6
7
8
9
10


action 0
action 1
action 2
action 3
action 0 under merchant
action 1 under merchant
action 2 under merchant
action 3 under merchant
total active days
recent active days

(2) python code (used for calculating the probability of prediction in three models)

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-

#%%
import pandas as pd
import numpy as np
import math
from sklearn.model_selection import KFold
#%%
data  =  pd.read_csv('data_final_f.csv') # loading the target data
data = data[pd.notnull(data['user_id'])]
data  = data.drop(['user_id','total','in_total'], axis = 1)
data_label = data['label']
data_no_label = data.drop(['label'],axis = 1)
#%%
# 6-fold cross validation
kf = KFold(n_splits = 6)
kf.get_n_splits(data)

fold_index = list(kf.split(data))

def get_index(fold_index, i_th_fold, train):
            train_index = fold_index[i_th_fold][0]
            test_index = fold_index[i_th_fold][1]
            if train == 'True':
                return train_index
            elif train == 'False':
                return test_index

train_index = []
test_index = []

for i in range(0,6):
    train_index.append(get_index(fold_index,i,train = 'True'))
    test_index.append(get_index(fold_index,i, train= 'False'))
#%%
# feature selection
feature_means = data.groupby('label').aggregate(np.mean).reset_index()
def feature_score(current, next):
     return abs(math.log2(current)- math.log2(next))
#%%
feature_name = ['number_of_0','number_of_1','number_of_2',
'number_of_3','in_number_of_0','in_number_of_1','in_number_of_2',
'in_number_of_3','total_active_days','recent_active_days']

feature_score_list = []

for i in range(0,len(feature_name)):
    score =
feature_score(feature_means[feature_name[i]][0],feature_means[feature_name[i]
][1])
    feature_score_list.append(score)

feature_score_dict = {}
```

```
for i in range(0, len(feature_name)):
    feature_score_dict[feature_name[i]] = feature_score_list[i]


feature_select = sorted(feature_score_dict, key=feature_score_dict.get,
reverse = True)[:6]

train_data = []
test_data = []
for i in range(0,6):
     train_data.append(data.loc[train_index[i]][feature_select])
     test_data.append(data.loc[test_index[i]][feature_select])
#%%
from sklearn.neighbors import KNeighborsClassifier as knn

def knn_model_proba(fold_number):
    knn_model = knn(n_neighbors = 10)
    knn_model.fit(train_data[fold_number],
data_label.iloc[train_index[fold_number]])
    knn_model.score(test_data[fold_number],
data_label.iloc[test_index[fold_number]])
    knn_pro_p_list = []
    for knn_pro_n, knn_pro_p in
knn_model.predict_proba(test_data[fold_number]):
        knn_pro_p_list.append(knn_pro_p)
    return(knn_pro_p_list)


#%%  #knn k selection
def knn_model_selection(fold_number,k):
    score_list =[]
    for i in range(1,k+1):
        knn_model = knn(n_neighbors = i)
        knn_model.fit(train_data[fold_number],
data_label.iloc[train_index[fold_number]])
        score_ = knn_model.score(test_data[fold_number],
data_label.iloc[test_index[fold_number]])
        score_list.append(score_)

    return(score_list)
#%%

# Logistic regression
from sklearn.linear_model import LogisticRegression as logit

def logit_model_proba(fold_number):
    logit_model = logit()
    logit_model = logit_model.fit(train_data[fold_number],
data_label.iloc[train_index[fold_number]])
    logit_model.score(test_data[fold_number],
data_label.iloc[test_index[fold_number]])
    logit_pro_p_list = []
    for logit_pro_n, logit_pro_p in
logit_model.predict_proba(test_data[fold_number]):
        logit_pro_p_list.append(logit_pro_p)
    return(logit_pro_p_list)
#%%
```

```
# qda
from sklearn.discriminant_analysis import QuadraticDiscriminantAnalysis as
qda

def qda_model_proba(fold_number):
    qda_model = qda()
    qda_model.fit(train_data[fold_number],
data_label.iloc[train_index[fold_number]])
    qda_model.score(test_data[fold_number],
data_label.iloc[test_index[fold_number]])
    qda_pro_p_list = []
    for qda_pro_n, qda_pro_p in
qda_model.predict_proba(test_data[fold_number]):
        qda_pro_p_list.append(qda_pro_p)
    return(qda_pro_p_list)


#%%
result_1 = pd.DataFrame(
         {'knn_pro_0': knn_model_proba(0),
         'logit_pro_0' :logit_model_proba(0),
        'qda_pro_0' :qda_model_proba(0),
         'label': data_label.iloc[test_index[0]]
          })

result_1.to_csv('result_1.csv', index = False)

result_2 = pd.DataFrame(
         {'knn_pro_1': knn_model_proba(1),
         'logit_pro_1' :logit_model_proba(1),
        'qda_pro_1' :qda_model_proba(1),
         'label': data_label.iloc[test_index[1]]
          })

result_2.to_csv('result_2.csv', index = False)

result_3 = pd.DataFrame(
         {'knn_pro_2': knn_model_proba(2),
         'logit_pro_2' :logit_model_proba(2),
        'qda_pro_2' :qda_model_proba(2),
         'label': data_label.iloc[test_index[2]]
          })

result_3.to_csv('result_3.csv', index = False)

result_4 = pd.DataFrame(
         {'knn_pro_3': knn_model_proba(3),
         'logit_pro_3' :logit_model_proba(3),
        'qda_pro_3' :qda_model_proba(3),
         'label': data_label.iloc[test_index[3]]
          })

result_4.to_csv('result_4.csv', index = False)

result_5 = pd.DataFrame(
         {'knn_pro_4': knn_model_proba(4),
         'logit_pro_4' :logit_model_proba(4),
```

```
        'qda_pro_4' :qda_model_proba(4),
         'label': data_label.iloc[test_index[4]]
          })

result_5.to_csv('result_5.csv', index = False)

result_6 = pd.DataFrame(
         {'knn_pro_5': knn_model_proba(5),
         'logit_pro_5' :logit_model_proba(5),
        'qda_pro_5' :qda_model_proba(5),
         'label': data_label.iloc[test_index[5]]
          })

result_6.to_csv('result_6.csv', index = False)

#================================================================================
==
# #%%
# name_dic = {}
#
# number_list = ['0','1','2','3','4','5']
# label_str = 'label'
# for i in range(0,6):
#     knn_pro_str = 'knn_pro_' + number_list[i]
#     logit_pro_str = 'logit_pro_' + number_list[i]
#     qda_pro_str = 'logit_pro_' + number_list[i]
#     name_dic[knn_pro_str] = knn_model_proba(i)
#     name_dic[logit_pro_str] = logit_model_proba(i)
#     name_dic[qda_pro_str] = qda_model_proba(i)
#     name_dic[label_str] = data_label.iloc[test_index[i]]
#
# result = pd.DataFrame.from_dict(name_dic)
#================================================================================
==
```

R code (used to plot the result , R markdown format)

```
---
title: "assignment"
author: "Chaomin Yuan"
date: "October 17, 2016"
output: html_document
---

```{r setup, include=FALSE}
knitr::opts_chunk$set(echo = TRUE)
```

loading package
```{r}
library(caret)
library(mlbench)
library(class)
library(e1071)
library(KernSmooth)
```
```

```
# loading propability
```{r}
setwd("C:/Users/Chaomin/Desktop/master 1/STAT5003/play_ground")
data <- read.csv("result_1.csv",header = TRUE)
truth <- data$label
```

#knn
```{r}
knn_sen_result <- c()
knn_spe_result <- c()
knn_F1_result <- c()
knn_AC_result <- c()

knn.TP <- knn.TN <- knn.FP <- knn.FN <- c()


x_knn <- seq(0,1,0.001)

knn.prob <- data$knn_pro_0

for ( i in 1:length(x_knn)) {

knn.preds <-  ifelse(knn.prob >= x_knn[i], 1, 0)

knn.TP <- c(knn.TP, sum((truth == knn.preds)[truth == "1"]))
knn.TN <- c(knn.TN, sum((truth == knn.preds)[truth == "0"]))
knn.FP <- c(knn.FP, sum((truth != knn.preds)[truth == "0"]))
knn.FN <- c(knn.FN, sum((truth != knn.preds)[truth == "1"]))

}
```

knn_roc_curve

```{r}
knn_roc_data <- data.frame(knn.TP, knn.TN, knn.FP, knn.FN)
knn_roc_data$Sen <- knn_roc_data$knn.TP/(knn_roc_data$knn.TP +
knn_roc_data$knn.FN)
knn_roc_data$Spe <- knn_roc_data$knn.TN/(knn_roc_data$knn.TN +
knn_roc_data$knn.FP)

#fit_knn <- locpoly((1-knn_roc_data$Spe), knn_roc_data$Sen, kernel = "normal",
bandwidth = 0.035)
#fit_knn.fn <- approxfun(fit_knn$x, fit_knn$y)

plot((1 - knn_roc_data$Spe),knn_roc_data$Sen, type = "o", col = 'blue', lwd =
2,main ='group1: knn ROC curve', xlab = 'False positive rate ',ylab = 'True
positive rate', ylim = seq(0,1), xlim = seq(0,1))
lines(x_knn,x_knn, lwd = 2, lty = 2, col = rainbow(5))
#lines(fit_knn, lwd = 2, col = 'lightpink')
legend(0.5,0.4,c('knn roc curve AUC score = 0.36','smooth line of roc curve',
'random guess line: AUC score = 0.5'),box.lwd=0.001,lwd=2,lty = c(1,1,2),col
= c('blue','lightpink','red'))

 # mc.x_knn <- runif(10000, min=0.3, max=1)
```
```

```
 #
 # max.y_knn <- max(fit_knn.fn(mc.x_knn))
 # max.y_knn
 # mc.y_knn <- runif(10000, min=0, max=max.y_knn)
 # area.ratio_knn <- sum(mc.y_knn < fit_knn.fn(mc.x_knn)) / 10000
 # auc_knn <- area.ratio_knn * 0.7 * max.y_knn
 # auc_knn
```

```
##knn
```{r}
logit_sen_result <- c()
logit_spe_result <- c()
logit_F1_result <- c()
logit_AC_result <- c()

logit.TP <- logit.TN <- logit.FP <- logit.FN <- c()

x_logit = seq(0,1,0.001)


logit.predict <- data$logit_pro_0

for (i in 1:length(x_logit)) {
logit.preds <- ifelse(logit.predict >= x_logit[i], 1, 0) #making predictions
with logistic model
logit.TP <- c(logit.TP, sum((truth == logit.preds)[truth == "1"]))
logit.TN <- c(logit.TN, sum((truth == logit.preds)[truth == "0"]))
logit.FP <- c(logit.FP, sum((truth != logit.preds)[truth == "0"]))
logit.FN <- c(logit.FN, sum((truth != logit.preds)[truth == "1"]))
}
```


logit_roc_curve
```{r}
logit_roc_data <- data.frame(logit.TP, logit.TN, logit.FP, logit.FN)
logit_roc_data$Sen <- logit_roc_data$logit.TP/(logit_roc_data$logit.TP +
logit_roc_data$logit.FN)
logit_roc_data$Spe <- logit_roc_data$logit.TN/(logit_roc_data$logit.TN +
logit_roc_data$logit.FP)

fit_logit_roc <- locpoly((1-logit_roc_data$Spe), logit_roc_data$Sen, kernel =
"normal", bandwidth = 0.02)
fit_logit_roc.fn <- approxfun(fit_logit_roc$x, fit_logit_roc$y)

plot((1 - logit_roc_data$Spe),logit_roc_data$Sen, type = "o", col = 'blue',
lwd = 2,main ='group1: logit ROC curve', xlab = 'False negtive rate ',ylab =
'True positive rate', ylim = seq(0,1), xlim = seq(0,1))
lines(x_logit,x_logit, lwd = 2, lty = 2, col = rainbow(5))
lines(fit_logit_roc, lwd = 2, col = 'lightpink')
legend(0.5,0.4,c('knn roc curve AUC score = 0.641','smooth line of roc curve',
'random guess line: AUC score = 0.5'),box.lwd=0.001,lwd=2,lty = c(1,1,2),col
= c('blue','lightpink','red'))

mc.x_logit <- runif(10000, min=0.01, max=1)
```
```

```
max.y_logit <- max(fit_logit_roc.fn(mc.x_logit))
max.y_logit
mc.y_logit <- runif(10000, min=0, max=max.y_logit)
area.ratio_logit <- sum(mc.y_logit < fit_logit_roc.fn(mc.x_logit)) / 10000
auc_logit <- area.ratio_logit * 1 * max.y_logit
auc_logit
```

lda

```{r}
lda_sen_result <- c()
lda_spe_result <- c()
lda_F1_result <- c()
lda_AC_result <- c()

lda.TP <- lda.TN <- lda.FP <- lda.FN <- c()

x_lda = seq(0,1,0.001)


lda.predict <- data$lda_pro_0

for (i in 1:length(x_lda)) {
lda.preds <- ifelse(lda.predict >= x_lda[i], 1, 0) #making predictions with
logistic model
lda.TP <- c(lda.TP, sum((truth == lda.preds)[truth == "1"]))
lda.TN <- c(lda.TN, sum((truth == lda.preds)[truth == "0"]))
lda.FP <- c(lda.FP, sum((truth != lda.preds)[truth == "0"]))
lda.FN <- c(lda.FN, sum((truth != lda.preds)[truth == "1"]))
}
```

lda roc curve

```{r}
lda_roc_data <- data.frame(lda.TP, lda.TN, lda.FP, lda.FN)
lda_roc_data$Sen <- lda_roc_data$lda.TP/(lda_roc_data$lda.TP +
lda_roc_data$lda.FN)
lda_roc_data$Spe <- lda_roc_data$lda.TN/(lda_roc_data$lda.TN +
lda_roc_data$lda.FP)

fit_lda_roc <- locpoly((1-lda_roc_data$Spe), lda_roc_data$Sen, kernel =
"normal", bandwidth = 0.02)
fit_lda_roc.fn <- approxfun(fit_lda_roc$x, fit_lda_roc$y)

plot((1 - lda_roc_data$Spe),lda_roc_data$Sen, type = "o", col = 'blue', lwd =
2,main ='group1: lda ROC curve', xlab = 'False negtive rate ',ylab = 'True
positive rate', ylim = seq(0,1), xlim = seq(0,1))
lines(x_lda,x_lda, lwd = 2, lty = 2, col = rainbow(5))
lines(fit_lda_roc, lwd = 2, col = 'lightpink')
legend(0.5,0.4,c('knn roc curve AUC score = 0.64','smooth line of roc curve',
'random guess line: AUC score = 0.5'),box.lwd=0.001,lwd=2,lty = c(1,1,2),col
= c('blue','lightpink','red'))
```

```
mc.x_lda <- runif(10000, min=0.01, max=1)
max.y_lda <- max(fit_lda_roc.fn(mc.x_lda))
max.y_lda
mc.y_lda <- runif(10000, min=0, max=max.y_lda)
area.ratio_lda <- sum(mc.y_lda < fit_lda_roc.fn(mc.x_lda)) / 10000
auc_lda <- area.ratio_lda * 1 * max.y_lda
auc_lda
```

plot three curve together

```{r}
plot((1 - knn_roc_data$Spe),knn_roc_data$Sen, type = "o", col = 'blue', lwd = 2, ylim = seq(0,1), xlim = seq(0,1),axes = FALSE , xlab = '', ylab = '')
par(new = TRUE)
plot((1 - logit_roc_data$Spe),logit_roc_data$Sen, type = "o", col = 'green', lwd = 2, ylim = seq(0,1), xlim = seq(0,1),axes = FALSE,xlab = '', ylab = '')
par(new = TRUE)
plot((1 - lda_roc_data$Spe),lda_roc_data$Sen, type = "o", col = 'gold', lwd = 2, ylim = seq(0,1), xlim = seq(0,1), xlab = "False positive rate(fall-out score: 1- Specificity)", ylab = "True positive rate(Sensitive socre)", main = 'group1: Comparision between three classifier: based on ROC curve')
lines(x_logit,x_logit, lwd = 2, lty = 2, col = rainbow(5))
legend(0.4,0.3,c('logistic classifier:AUC score = 0.662','knn classifier: AUC score:0.36','lda classifier:AUC score = 0.45', 'random guess line: AUC score = 0.5'),box.lwd=0.001,lwd=2,lty = c(1,1,1,2),col = c('green','blue','gold', 'red'))
```

TEAM TASK MEETING AGENDA

TEAM MEETING AGENDA
____Group 18____
Meeting to be held _____at ABS level 3___

_____29/09/2016_____
_____15:15PM_____
Chairperson: _____MENGDIE XU_____
Minute-Taker: _____CHAOMIN YUAN_____

1. Apologies: none
2. Confirmation of agenda (5 minutes)(Chair)
3. Confirmation of minutes of none (0 minutes)(Chair)
4. Business arising from minutes of (none) (0 minutes)(Chair)
5. Items (30 minutes)(Chair)
   (1) Team member introduction
   (2) Project requirement discussion
   (3) Topic browsing and discussion
   (4) Research work division e.g. data available, relevant academic research, etc.
6. Any other business (0 minutes)(Chair)
7. Forward agenda items (5 minutes)(Chair)
   (1) Research results discussion
   (2) Topic decision
8. Next meeting (Chair)
   06/10/2016 at ABS level 3

TEAM MEETING AGENDA

____Group 18____

Meeting to be held _____at ABS level 3___

_____06/10/2016_____

_____15:15PM_____

Chairperson: _____CHAOMIN YUAN_____

Minute-Taker: _____ MENGDIE XU _____

1. Apologies: none
2. Confirmation of agenda (5 minutes)(Chair)
3. Confirmation of minutes of 29/09/2016 (3 minutes)(Chair)
4. Business arising from minutes of (none) (0 minutes)(Chair)
5. Items (30 minutes)(Chair)
   (1) Research results discussion
   (2) Topic and data selection
   (3) Model discussion
   (4) Work division e.g. topic background introduction, data exploration, model computing etc.
6. Any other business (0 minutes)(Chair)
7. Forward agenda items (5 minutes)(Chair)
   (1) Model selection
8. Next meeting (Chair)
   13/10/2016 at ABS level 3

TEAM MEETING AGENDA

____Group 18____

Meeting to be held _____at ABS level 3___

_____13/10/2016_____

_____15:15PM_____

Chairperson: _____ZHONGSHU LIN_____

Minute-Taker: _____ BOFAN DONG _____

1. Apologies: none
2. Confirmation of agenda (5 minutes)(Chair)
3. Confirmation of minutes of 06/10/2016 (3 minutes)(Chair)

4. Business arising from minutes of (none) (0 minutes)(Chair)
5. Items (30 minutes)(Chair)
   (1) Data exploration and model computing results discussion
   (2) Model selection
   (4) Discussion on possible limitations
   (5) Report structuring and work division
6. Any other business (0 minutes)(Chair)
7. Forward agenda items (5 minutes)(Chair)
   (1) Report finalizing
8. Next meeting (Chair)
   20/10/2016 at ABS level 3

TEAM MEETING AGENDA
____Group 18____
Meeting to be held _____at ABS level 3___

_____20/10/2016_____
_____15:15PM_____
Chairperson: _____Mandy_____
Minute-Taker: _____ BOFAN DONG _____

9. Apologies: none
10. Confirmation of agenda (5 minutes)(Chair)
11. Confirmation of minutes of 13/10/2016 (3 minutes)(Chair)
12. Business arising from minutes of (none) (0 minutes)(Chair)
13. Items (30 minutes)(Chair)
    (1) Report finalizing
14. Any other business (0 minutes)(Chair)
15. Forward agenda items (0 minutes)(Chair)
16. Next meeting (Chair)
    None

THE UNIVERSITY OF SYDNEY

Minutes of meeting for _____ Group 18_____

Date: _____29/09/2016_____ Time: _____15:15PM_____ Location: _____ABS level 3_____

Chairperson: _____MENGDIE XU_____

Minute-Taker: _____CHAOMIN YUAN_____

Document tabled: _____none_____

Present: _____none_____

Apologies: ____none_____

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
(1) Team member introduction	*Ice-breaking games *	Games	All members	15:15-15:20	Casual
(2) Project requirement discussion	* * * * *	Discussion	All members	15:20-15:25	Free discussion
(3) Topic browsing and discussion	*Website and data library *	Research	All members	15:25-15:40	Leaded discussion
(4) Research work division e.g. data available, relevant academic research, etc.	*YUAN&XU data research *The rest academic research *	Discussion	All members	15:40-15:45	Free discussion

Souce: TAFE Access Division "Communication for Business", 2000

Date: _____06/10/2016_____ Time: _____15:15PM_____ Location: _____ABS level 3_____

Chairperson: _____ CHAOMIN YUAN _____

Minute-Taker: _____MENGDIE XU_____

Document tabled: _____none_____

Present: _____none_____

Apologies: ____none_____

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
(1) Research results discussion	*Data mining in ecommerce *Predictive analysis	Discussion	All members	15:15-15:25	Free discussion
(2) Topic and data selection	*Online retailing and customer features	Discussion	All members	15:25-15:30	Free discussion
(3) Model discussion	*Classification KNN Bayesian Logistic etc	Discussion	All members	15:30-15:40	Leaded discussion
(4) Work division	*YUAN&XU model computing *The rest topic background research	Discussion	All members	15:40-15:45	Leaded discussion

Souce: TAFE Access Division “Communication for Business”, 2000

Date: _____13/10/2016_____ Time: _____15:15PM_____ Location: _____ABS level 3_____

Chairperson: _____ ZHONGSHU LIN _____

Minute-Taker: _____ BOFAN DONG _____

Document tabled: _____none_____

Present: _____none_____

Apologies: ____none_____

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
(1) Data exploration and model computing results discussion	*Normal VS non-normal *Model performance	Discussion Discussion	All members	15:15-15:30	Leaded discussion
(2) Model selection	*Benchmark KNN, Bayesian naïve, and Logistic	Discussion	All members	15:30-15:40	Free discussion
(3) Discussion on possible limitations	*Underlying assumptions	Discussion	All members	15:40-15:50	Free discussion
(4) Report structuring and work division	*YUAN&XU statistical presentation *The rest report structuring	Discussion	All members	15:50-15:55	Leaded discussion

Souce: TAFE Access Division "Communication for Business", 2000

Date: _____20/10/2016_____ Time: _____15:15PM_____ Location: _____ABS level 3_____

Chairperson: _____ Mandy _____

Minute-Taker: _____ BOFAN DONG _____

Document tabled: _____none_____

Present: _____none_____

Apologies: ____none_____

Agenda Item	Key Points	Action	By Whom	When	Communication Strategy
Report finalizing	*Proofing and referencing *Presentation *Grammar check	Editing and discussion	All members	15:15-15:45	Leaded discussion

Souce: TAFE Access Division “Communication for Business”, 2000