

# Individual Final Report

Ya Liu

## Introduction

Our project is to predict whether a user will buy a product in the next order. Our group has three members. In summary, one is responsible of exploratory data analysis; one is responsible of feature engineering, and I am responsible of modeling.

## My Responsibility

Specifically, my work during this project includes merging all features we create with the main dataset provided by Kaggle, and then develop models. The code I contribute is from ‘add features into train dataset’ part to the end. For the final report, I am responsible of the contents from part 5 to part 11 except the theoretical part of logistic regression. Besides, I design PowerPoint for presentation.


## Results

The dataset after merging is shown below. Here we have 14 features and one target.

| user_id | product_id | times_bought_up | reorder_ratio_up | ratio_last4_orders_up | n_orders_users | n_products_users | avg_products_users | dow_most_user | hod_most_user | reorder_ratio_user | shopping_freq | times_bought_prod | reorder_ratio_prod | position_cart_prod | reorder_ratio_dept | reordered |     |
|---------|------------|-----------------|------------------|-----------------------|----------------|------------------|--------------------|---------------|---------------|--------------------|---------------|-------------------|--------------------|--------------------|--------------------|-----------|-----|
| 0       | 1          | 196             | 10               | 1.000000              | 1.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 35791              | 0.776480           | 3.721774           | 0.471714  | 1.0 |
| 1       | 1          | 10258           | 9                | 1.000000              | 1.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 1946               | 0.713772           | 4.277492           | 0.438319  | 1.0 |
| 2       | 1          | 10326           | 1                | 0.166667              | 0.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 5526               | 0.652009           | 4.191097           | 0.412660  | 0.0 |
| 3       | 1          | 12427           | 10               | 1.000000              | 1.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 6476               | 0.740735           | 4.760037           | 0.438319  | 0.0 |
| 4       | 1          | 13032           | 3                | 0.333333              | 0.50           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 3751               | 0.657158           | 5.622767           | 0.466878  | 1.0 |
| 5       | 1          | 13176           | 2                | 0.222222              | 0.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 379450             | 0.832555           | 5.095947           | 0.412660  | 0.0 |
| 6       | 1          | 14084           | 1                | 0.100000              | 0.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 15935              | 0.810982           | 5.792595           | 0.505622  | 0.0 |
| 7       | 1          | 17122           | 1                | 0.166667              | 0.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 13880              | 0.675576           | 6.257421           | 0.412660  | 0.0 |
| 8       | 1          | 25133           | 8                | 1.000000              | 1.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 6196               | 0.740155           | 7.001614           | 0.505622  | 1.0 |
| 9       | 1          | 29388           | 2                | 0.200000              | 0.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 2523               | 0.539041           | 6.495838           | 0.438319  | 1.0 |
| 10      | 1          | 26405           | 2                | 0.200000              | 0.00           | 10               | 59                 | 5.900000      | 4             | 7                  | 0.694915      | 20.259259         | 1214               | 0.441516           | 3.116969           | 0.250641  | 1.0 |

Then I did oversampling to deal with the imbalance of the dataset. We can see from the figure below that the frequency of class 0 is the same as that of class 1.

| target | frequency |
|--------|-----------|
| 0      | 2,334,883 |
| 1      | 253,930   |



| target | frequency |
|--------|-----------|
| 0      | 2,334,883 |
| 1      | 2,334,883 |

Then I spited the whole dataset into train and test dataset by proportion of 0.7 and 0.3 respectively.

Regarding the modeling, I developed classifiers including logistic regression, KNN, decision tree, random forest, Naïve Bayes, SVM. The best result comes from random forest with the number of trees in the random forest being 100. We can see from the classification report that the accuracy and ROC score is pretty high and also the confusion matrix indicates it predicts well in

the testing dataset. The feature importance plot below shows the features that influence the model most.

```

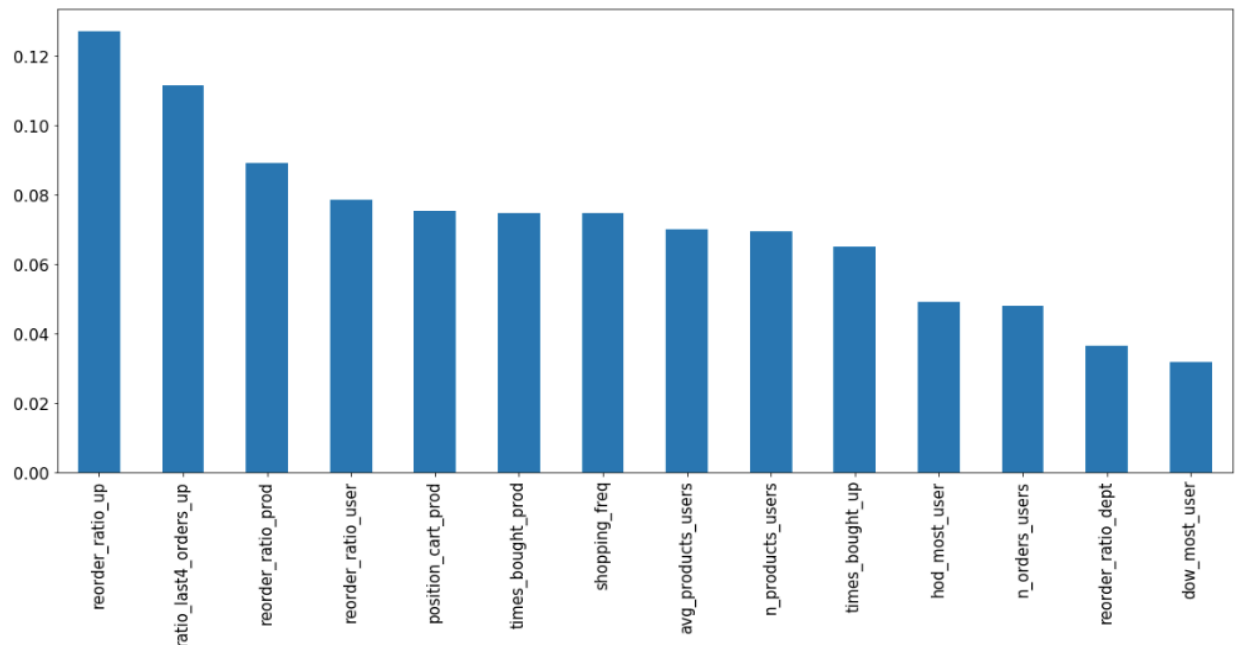
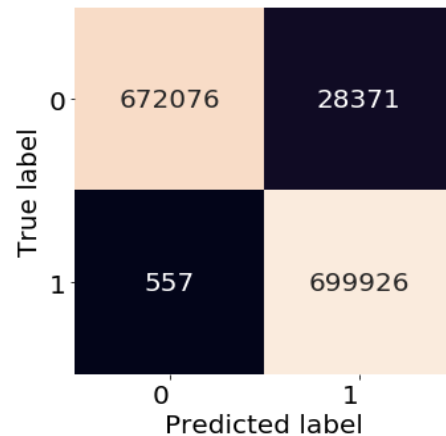
Classification Report:
              precision    recall  f1-score   support

     0.0         1.00      0.96      0.98     700447
     1.0         0.96      1.00      0.98     700483

 accuracy          0.98
  macro avg       0.98      0.98      0.98
 weighted avg     0.98      0.98      0.98
  
```

Accuracy : 97.93508597859993

ROC\_AUC : 99.97318652685728



## Summary and Conclusion

In summary, what I contribute in this project is merging all features into one dataset and modeling. I tried several classifiers. Some of them predict not as good as random forest. SVM takes too long to run out. It turns out that random forest has the best prediction with misclassification rate being 2.06%.

There are some improvements I can make probably to get a better model. First, I use default value in the random forest. Specifically, I choose the number of trees in the random forest being 100, which can be improved by hyperparameter tuning. By finding the optimal hyperparameters, the model may be improved. Second, I could try to use gradient boosting instead of bagging in random forest, which may also give a boost to our model.

## Percentage of the code

There are 15 lines of code that I found in Internet and modified them, and there are about 150 lines of code that I found in the lecture code and modified some lines. So the percentage of code that I found from the internet would approximately be  $\frac{135}{15+150} = 81.82\%$ .

## Reference

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*. Vol. 1. No. 10. New York: Springer series in statistics, 2001.