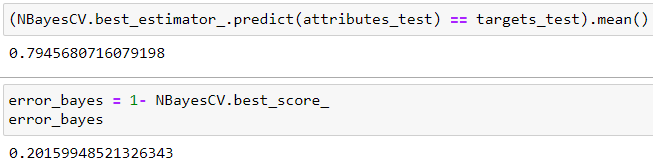
Week 7 Report Doris Chen

This week our team meet for 2 hours yesterday. Since last the week we did feature extraction works, features are extracted into the seller information table, user seller information table (with corresponding repeat buyer info: Label: 0 is not a repeat buyer and 1 is a repeat buyer for a unique user & a unique seller), user action table, etc. Thus, for this week firstly we merged all feature tables together on user ID and seller ID into a big table, and we decided to take other entries except Label as attributes, so Label is the target of prediction. (We also one hot encoded user action because we only care if the user did the action rather than the action count).

During the meeting, I focused on Naïve Bayes prediction. As Dr. Shibberu said in class, Naïve Bayes is calculating the conditional probability of a class label given a data sample. For our model, we are given the P(features|label) and predict P(label|features), which we are predicting if the user will become a repeat buyer given information in attributes data.



This is the result I got for Bayes prediction, and the accuracy rate for Naïve Bayes is not as high as decision tree and logistic regression, but actually, we got a more than 90% prediction baseline because of biased data. Thus, the Bayes model is not appropriate for our prediction. We also use Cross-Validation which the validation data is a subset of original data and only used for testing as we are not given testing data.

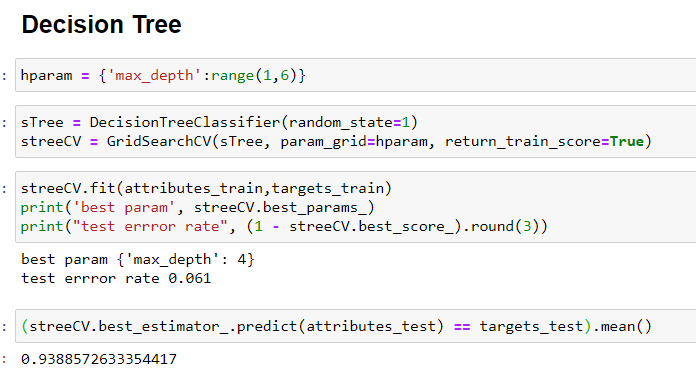
I think the next step we are going to solve is to eliminate the bias in our training data and try more models on our prediction to see if the accuracy can be improved.

Week 8 Report Doris Chen

Total Working Hour: 4 hrs

This week, each of us did some research on solving last week’s problem (unbalanced data) and studying new prediction algorithms for 2 hours, and we tried to increase the size of the label=1 data by copying the data. However, the method did not make so much sense. Thus, we decided to redo our prediction after standardizing our data.

Then, we met for 2 hours to try several different classifiers. Because some negative entries existed after standardization, the Naïve Bayes did not work, so we also tried Random Forest. I mainly worked on the Decision Tree, and I got the result with 93.88% accuracy and max depth of 4 which is the same as what we got in last week:

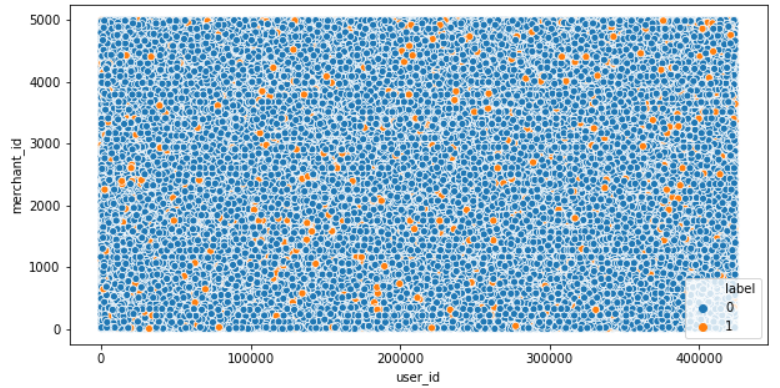


After data processing, we got some new feature data, so we also did some feature extraction and visualization together.

Week 8 Report Doris Chen

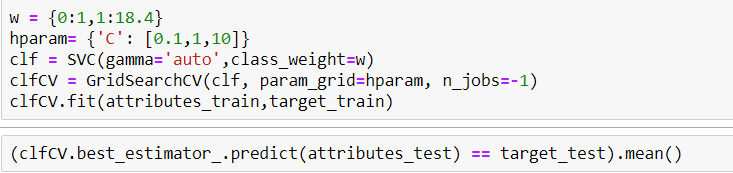
Total Working Hour: 4 hrs

This week, I did some research on solving the unbalanced data and studied SVM for 2 hours first.



As you can see, there are much more 0s than 1s. So, the accuracy of more than 90% can be achieved simply by predicting the complete set as majority label i.e. 0. But that is no help since we are building a classifier to classify minority labels (important) from the majority.

Then, we met for 2 hours to apply class-weights in accordance with the class distribution. Class-weights is the extent to which the algorithm is punished for any wrong prediction of that class. After creating our new test and train data, we tried to predict several weighted algorithms. I mainly focused on Support Vector Classification.



However, I tried to run the model both on my computer and on the Gauss, but it was like an endless loop that takes forever. Thus, I found out several solutions which I can try next week from the internet:

1. Reducing training set size.
2. Reducing dimensionality.
3. Parameters. Use SVC(probability=False) unless you need the probabilities.
4. Different classifier.

As you know our group is working on a very imbalance dataset in which the number of targets equal to 0 is more than 15 times the number of targets equal to 1, so the accuracy of bigger than 90% can be achieved simply by predicting the majority label (i.e. 0.) But it is kind of meaningless. We tried several data augmentation methods in last two weeks, but we finally solved this problem by adding class-weights according to the class distribution in this week. Class-weights is the extent to which the algorithm is punished for any wrong prediction of that class. After trying several models with the weight, we got much lower accuracies around 55% on average, but this should be the real accuracy without overfitting. And next week, we are going to work and explore more on improving the prediction accuracy of our data.

Week 10 Report Doris Chen

Total Working Hour: 4 hrs

This week, I met with my group for 3 hours. Since we had a new phase of the project each week in the previous few weeks, we put codes for each phase into different python scripts. For example, our visualization, unweighted predictions, and weighted predictions were in different scripts, but it is difficult to see if the weighted method really improves our prediction accuracy by going back and forth through the scripts.

Moreover, last week I hard-coded where we got class-weights, and this week my teammates solved this problem by converting Series to Dictionary. Then, we cleaned up and merged our previous works this week, and we could now clearly see the difference between the weighted and unweighted methods through the printed errors and graphs.

In addition, we also discussed the PPT and final presentation, then we decided to separate our presentation into 5 parts. We also discussed which graphs are valuable that we can put in our PPT. I also worked on my own part of the presentation for 1 hour.