Each team is expected to submit one copy of their project report, code, and data.

(1) The report should be a pdf file.

(2) Include your team team, team member and UID , the role of each team member in the first page of the report.

(3) The report should be include sections 1) Introduction (show the background, motivation, challenge, contribution) 2)related work 3)notation and problem statement 4)your proposed model 5)Experiments (Include at least one competitors) 6) Conclusion. (7) **Detailed** description how to run your code.

(4) Code

(5) Data: If you can demonstrate the performance of the model using part of the entire data, you can just use part of data. CCLE only allow submissions under 100 MB. If you want to submit the entire data, you may want to submit the entire data in person by appointment.

(6) Make sure the code is run-able under linux or windows. Make it clear. What is the operating system, What is the software to run your code.

(7) We will strictly follow the descriptions in your report to run your code. Do not hardcode anything (for example, path). Write your descriptions as detailed as possible.

(8) No late submissions or drafts will be accepted

Project Report

Team Name: Observer

Team Member:

1) XIANG ZHONG 204412666

2) Yang Pei 304434922

3) Hongbo Zhao 604426609

4) Qianwen Zhang 004401414

5) Jing Zhao 404426610

6) Zhe Sun 604435430

Role of each member:

Sun Zhe:

Doing surveys about the algorithms and features and Exploitinghow to use those state of the art algorithms/packages/modular

Xiang Zhong:

Data Preprocessing and figure out how to select good figures

Yang Pei and Hongbo Zhao:

Implementing the algorithms, and do evaluation of different algorithms.

Qianwen Zhang and Jing Zhao:

Writing reports and prepare the presentations.

1. Introduction:

Background and Motivation:

In online advertising, click-through rate (CTR) is a very important metric for evaluating ad performance. The meaning of the click rate of advertising is to measure basically how much people have clicked the advertisements during a certain period. Obviously, more click means more money return for the advertisers. Therefore, as a result, click prediction systems are essential and widely used for sponsored search and real-time bidding. With a known click rate, the advertisers can try to adjust their advertisement in a more proper way.

1. Related Work:

In the paper ["A two-stage ensemble of diverse models for advertisement ranking in KDD Cup 2012." KDDCup (2012).] It surveys lots of algorithms and techniques, which can be used in, click rate prediction. However, because some of the important attributes provided in the training set and the testing set are numerical data instead of categorical data, while our training set and testing set only contains categorical data (we will discuss about it later), we are not able to directly adapt their approaches.

1. Problem Statement:

The objective of our project is to predict whether a mobile ads will be clicked or not, given some information (attributes) of the ads. The datasets we have are training data set, which contains ten days of information about the ads, and a testing data set, which contains one days of information about the ads.

The training data set is about 7 GB, which have nearly 20 million records, and testing data set is about 1.27 GB, which have nearly 5 million records for us to predict.

Inside the data set, there are some attributes are known, such as:

banner\_possite\_idsite\_domainsite\_categoryapp\_idapp\_domainapp\_categorydevice\_iddevice\_ipdevice\_modeldevice\_typedevice\_conn\_type.

However, there are also some attributes that we don’t know what are they.

And also, it should be noted that all the data or value of each attributes are categorical data, and all ads’ ids are unique in the dataset. Therefore, we don’t have any numerical data and cannot compute the click rate respected to each ads as many of the similar problem that used to predict click rate, which makes the problem becomes difficult and limited our model selection and etc.

Also, the number of values of the attributes varies from 4 to 10000, which are very different from each other.

The following is a short example of the dataset:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| id | click | C1 | C2 | site\_category |
| "1000009418151090000 | 0 | 1fbe01fe | f3845767 | 28905ebd |
| "10000169349117800000 | 0 | 1fbe01fe | f3845767 | 28905ebd |
| "10000371904215100000 | 1 | 1fbe01fe | f3845767 | 28905ebd |
| "10000640724480800000 | 1 | 1fbe01fe | f3845767 | 28905ebd |

Evaluation Metric:

In this problem, because what we can predict is the probability of whether ads will be click, we can use the AUC as the evaluation metric, which is also a well-defined metric to evaluate this kind of problems.

That means, bigger AUC, the better the performance of the prediction.

1. Attributes selection

Because the data set are so large, it is not possible for us to put all the attributes or the data for training and testing, we try to filter out some attributes that are not useful. Therefore, the main challenge here is how to find out those attributes that are not useful in classification in this problem.

The meaning of useful of an attribute is that, if some values have high positive correlation of clicks of ads, while some values have high negative correlation of clicks of ads, we should consider this kind of attribute is useful. Otherwise, the attribute is not useful and we should filter it out.

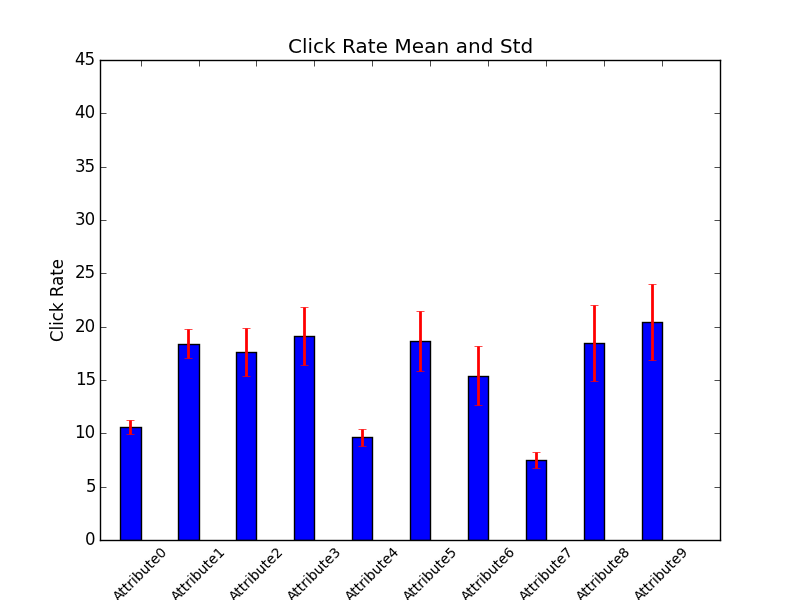
To find the correlation, we try to do some algorithm like frequent pattern mining, here we can think of it high variance pattern mining.

For a single attribute, we try to compute each value’s click rate (the meaning of click rate is how much click will appear if the value appears). If the variance is high, it means the attribute is useful since some values have high positive correlation of clicks of ads, while some values have high negative correlation of clicks of ads.

Beside just consider the relation between click and a single attribute, we also compute the relation between click and two attributes, for example, it makes sense that the ads’ site category combining the app’s category will have a strong relation with click of the ads.

However, because our limit of computing resource, we don’t consider the relation between click and more than 2 attributes, which I am sure it can produce more accurate results.

Here is the example of the variance and mean of the click rate of each attribute:

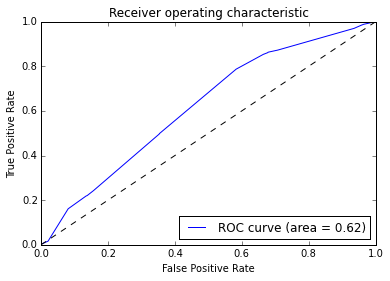


You can see that some attributes is good for classification, such as attribute 8 and attribute 9, while attribute 7 is bad.

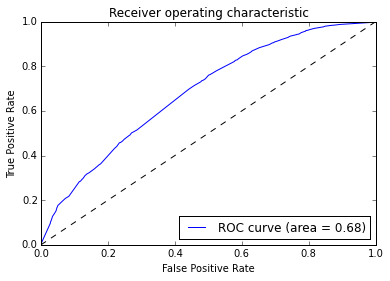
And the following is the improvement with and without attributes filtering.

(Using the ROC and AUC of logistic regression as examples)

Before filtering:



After filtering:



We can find that the performance of using attributes after filtering is better.

1. Proposed Model

4.1 Decision Tree

4.2 Random Forest

4.3 Naïve Bayesian Method

4.4 Logistic Regression

The logistic Regression algorithm is very useful in Classification if the given value of data set is numerical data. However, because the data set we have don’t have any numerical data. We have to find some way to transfer our categorical data into numerical data.

To do the transformation, we use the techniques of binary extraction and dimension increasing. That is, every dimension of the input to the logistic regression is a flag of whether a particular value of a particular attribute exists. For example, if the attribute app category has two values: game or tools, after the transformation, the input to the logistic regression will have two dimensions, one is whether the app is a game, and the other is whether the app is a tool. In this case, we are able to transfer the categorical data into numerical data, which can be able to apply to the logistic regression algorithm.

It should be noted that sometimes because of the number of values of all the attributes, the input will have too much dimension, which is hard for the algorithm running, that’s why the previous step attributes filtering is important. In addition, even though after filtering, the dimension is so high that other regression models such as SVM are difficult for our computing resources.

4.5 Neural Network

The Neural Network is the state of the art techniques for machine learning, however, we don’t prefer this kind of method in our problem, because using Neural Network means we would be able to know the inside logic of the problem. Also, it is extremely slow for our problem

4.6 Hybrid training algorithm

Beside just use the training algorithms list above, we also try to combine different algorithms hoping to boost the performance. Therefore, we try to combine the output of Decision Tree and Naïve Bayesian to be the input of logistic regression.

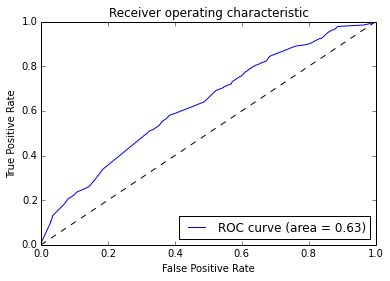
4.7 Novel Method

We proposed a novel method to classify the click through rate. Due to the key-value constraints of the data, we try to convert it to some real number for better classification. We assume that each attribution has some attraction to make them click the ad. We also assume that each attribution is independent of each other. Based on these two assumption, we quantization each attribution by the click rate in the training set. After the quantization, we use Gaussian Naive Bayes to fit the data and obtain a model. Then, during the test phase, we could not directly obtain the click rate for each value of the attribute. So, we use some procedure similar to EM algorithm. First, using the population of each value in an attribute as the estimate of the click rate, then we use the trained model to classify the data. After the classification, we update the click rate based on the result and do another iteration. We stop the iteration after some predefined steps.

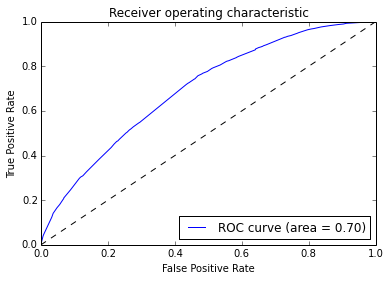
1. Experiments and Evaluation

Here we present the ROC of each model and the AUC summary table for you, therefore, you can observe which algorithm is more suitable for the problem.

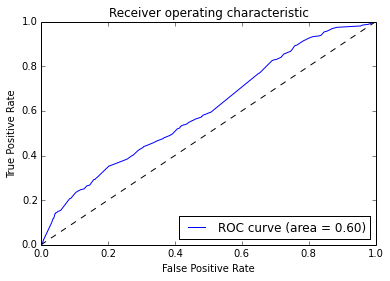
* Decision Tree:



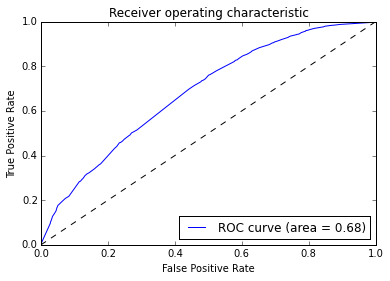
* Random Forest



* Naïve Bayesian:



* Logistic Regression:



The following is a table summary the performance of different models:

Neural Network:

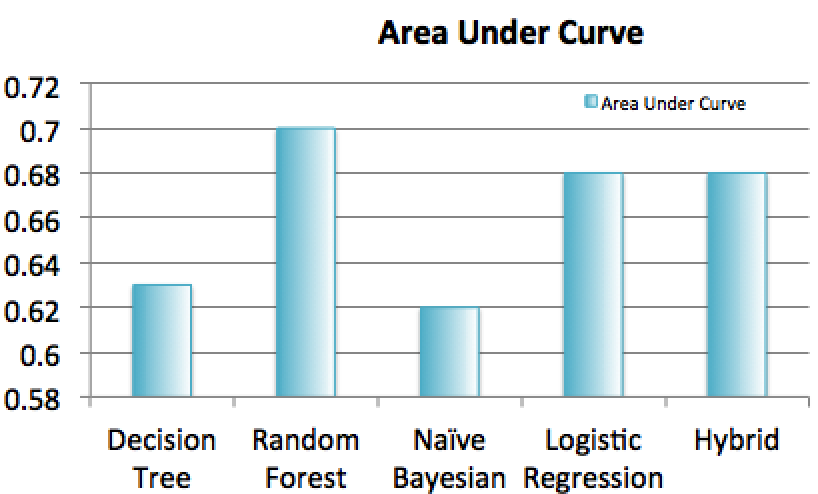
train error: 17.06000000% test error: 17.47137632%

Because we cannot extract the probability of click or not by using Neural Network, we cannot draw the AUC.

AUC table of different algorithms:

|  |  |
| --- | --- |
| Algorithm | AUC |
| Decision Tree | 0.63 |
| Random Forest | 0.7 |
| Naïve Bayesian | 0.62 |
| Logistic Regression | 0.68 |
| Hybrid | 0.68 |

And the histogram graph



As a result, we can find that Random Forest perform the best in our problem. This is consistent with our sense, because random forest is the best one to classify the categorical data, which is the only type of data we have.

1. Conclusion

6.1) Summary

In this project, we try to predict the click of advertisement exist on the mobile applications, given some categorical data describing attributes of other advertisement and their label (click or not).

Since the number of attributes and the size of dataset are too large to be used in training algorithms, we try to reduce the dimensions by attributes selecting, we create a technique called Effective Attribute Pattern Mining, which can be used to filter out non-use categorical data.

We have exposed several mainstream algorithms, including logistic regression, naïve Bayesian, random forest and neural network. Finally, we find that random forest has the best performance.

6.2) Future Work

The most challenging tasks here is that how to select better attributes and how to deal with the only categorical data or do we need to transfer these kinds of categorical data into numerical data? If so, how? And because of our limited computing resources, we can only afford training a relatively small subset of the training data, which may degrade the actually performance of the training algorithms.

Therefore, in the future work, we should try to find some more techniques to deal with these challenges.

1. Details on how to run the demo code

Here we provide some details about how to run our code, we have made a demo code for you, therefore, you can be more easily see how our project is going on.

1. Prerequisites: Here we use MacBook Pro, Mac OS to run the program, therefore, we will provide the steps that can run our demo code.

Because all our codes are run by ipython notebook, therefore, we should install ipython notebook first, and the module for python training algorithms too.

It is pretty easy:

* 1. Make sure you have install python version 2.7
  2. Install pip follow the steps in the website bellow: <https://pip.pypa.io/en/latest/installing.html#install-pip>
  3. Install ipython notebook: follow the steps in the website below:

<http://ipython.org/install.html>

* 1. Install the python machine learning module: scikit-learn:

Follow the link below:

<http://scikit-learn.org/stable/install.html>

* 1. Install pylab (which is used to draw the ROC graph):

Follow the link below:

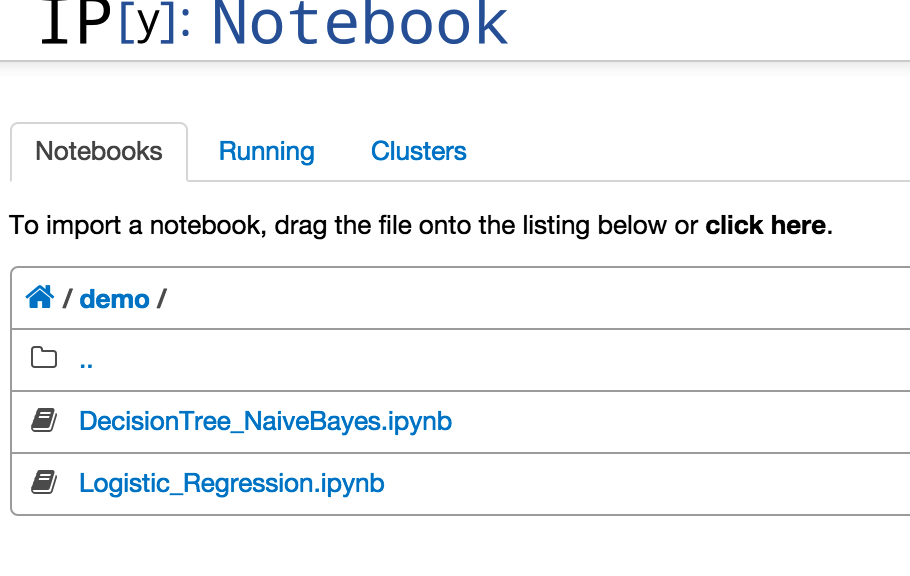
<http://matplotlib.org/users/installing.html>

f) Congratulation, you have installed all the necessary program to run our program, enjoy it!

1. Extract and open the “**Observer\_project\_code.zip”**, you will see many raw files and a directory which is **“demo/”** directory. The demo directory contains the files that are for you to run the program and see the result as quickly as possible.
2. Because the raw data is so big, here we provide the dataset that is already preprocessed, which is “train\_1234\_new\_rehashed.txt” (training set), “test\_6\_new\_rehashed.txt” (testing data)
3. To open the .ipynb file (which is our program), open the terminal, change the working directory to the “demo/” directory, and type the command:

“ipython notebook”, (make sure you are using Chrome browser!!)

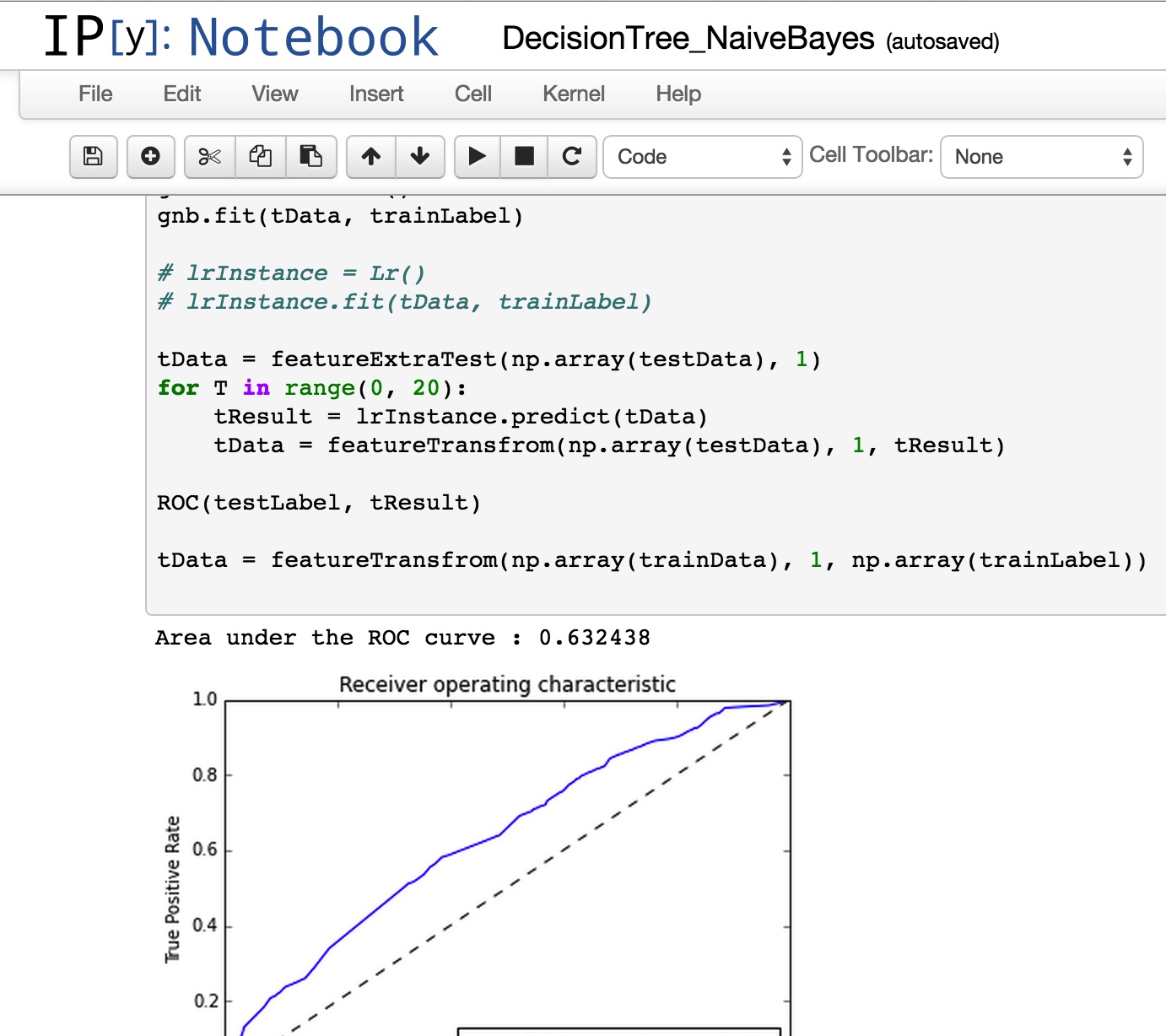
And what you will see is like this:



1. The running program of Models : Decision Tree, Naïve Bayesian and Random Forest are contained in the file: “ DecisionTree\_NaiveBayes\_RandomForest.ipynb ”

To run these model, just open the ipynb file in the demo/ directory, and run every block of the program, and you will see the result

The code is like this:



1. And the program of model Logistic Regression is contained in the file

“Logistic\_Regression.ipynb”, and run all the block of codes in the file and you will see the result.

6. If you are feeling confortable with the python code and ipython notebook, you can explore our raw codes, which are “.ipynb” files besides in the “demo/” directory in the “Observer\_project\_code/” directory.

If you come across any question or problem, feel free to tell me or email me:

[zhongxiang@g.ucla.edu](mailto:zhongxiang@g.ucla.edu). Thank you