Alibaba Advertisement CTR Algorithm Optimization

PSTAT 234 Final Project

Instructor: Prof. Sang-yun Oh
Author: Yiru Wang, Alex Wu, Sarah Li, Karry Deng

Agenda

- Abstract
- Data Introduction
- Regression Analysis
- Support Vector Machine
- Neural Network
- Conclusion and Discussion

Abstract

Data: 21st Century Gold

- Big data help companies with optimizing revenue
- Advertisement algorithm helps large shopping platforms with boosting customer usage
- Companies like Alibaba and ByteDance hold billions of customer data
    - These data further improves their company algorithm and increases their revenue
- CTR: Click Through Rate

Our Research Process

- Finding data
- Researching algorithm
- Building models
- Results analysis

Data Introduction

Data Source

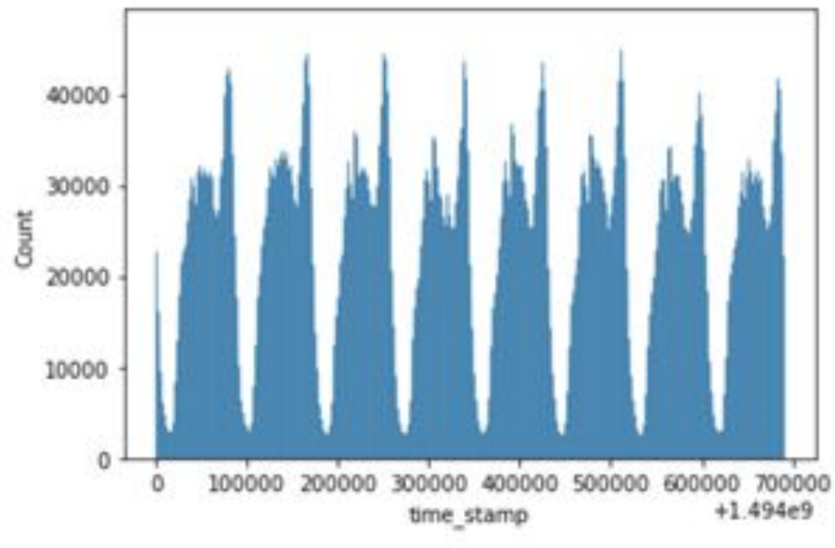
淘宝网
Taobao.com

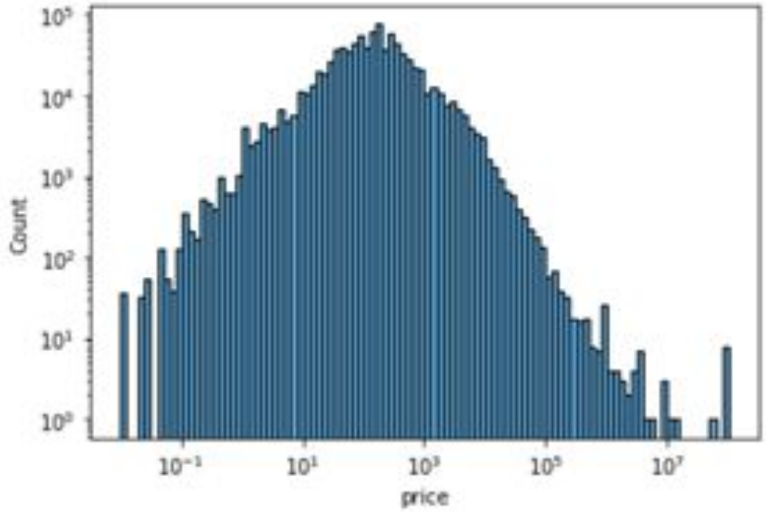
1.1 million users
26 million records

Table	Description	Feature
raw_sample	raw training samples	User ID, Ad ID, nonclk, clk, timestamp
ad_feature	Ad's basic information	Ad ID, campaign ID, Cate ID, Brand
user_profile	user profile	User ID, age, gender, etc
raw_behavior_log	User behavior log	User ID, btag, cate, brand, timestamp

[Tianchi: Data Sets (aliyun.com)]

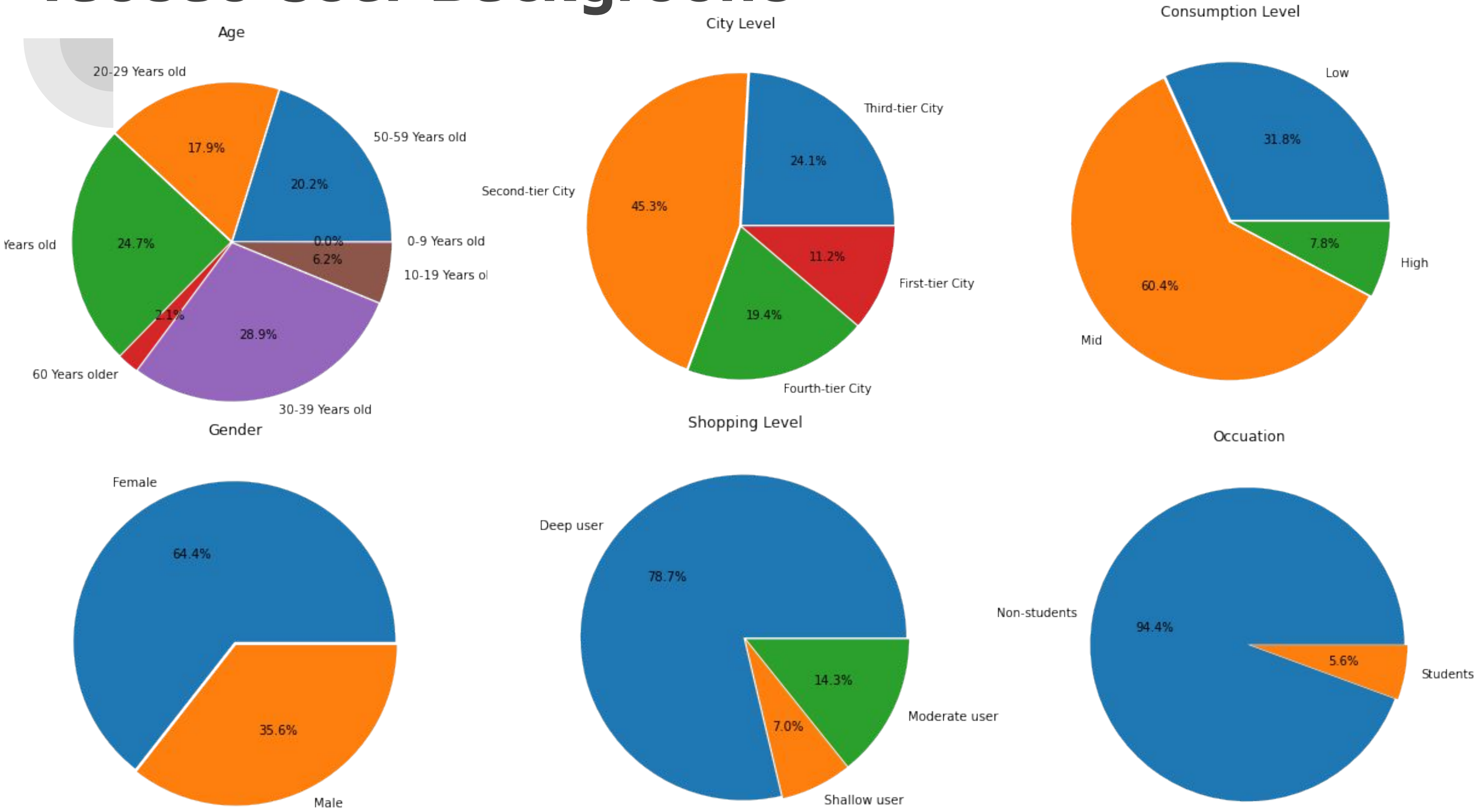
Clicks Time Distribution and Goods Price Distribution





Represent real world data

Taobao User Background



Regression Analysis

Logistic regression

We first started with a simple model: Logistic regression.

See how much signal we can pull out using basic method.

Logistic regression

- Y = click, X = features
- Training sets(75%), Testing sets(25%)

Logistic regression - Confusion matrix

Predicted Values

Actual Values

Test set N = 15000	Negative 0 unclick	Positive 1 click
Negative 0 unclick	TN 14117	FP 0
Positive 1 click	FN 883	TP 0

Logistic regression classified everything as unclick.
Logistic regression can not capture the click feature at all.

Logistic regression

Instead of predicting the click and nonclick, we tried to predict the probability of click.

```
: y_prob
```

	0	1
0	0.930223	0.069777
1	0.955861	0.044139
2	0.879269	0.120731
3	0.955928	0.044072
4	0.942675	0.057325
...	...	...
14995	0.940499	0.059501
14996	0.956010	0.043990
14997	0.960632	0.039368
14998	0.962795	0.037205
14999	0.966955	0.033045

The probability of clicking is very low which means the problem can not be solved by changing the threshold for logistic regression.

Need to apply another method.

Neural Network

Neural Network

We used the basic form of neural network which is a 3 layer fully-connected neural network, with ReLU activation function between each layer.

We tried different architectures such as different number of hidden layers, neurons, and learning rate.

Neural Network

We found that “model_3” which has 5 hidden layers with 15 neurons and 0.01 learning rate is the best model regarding to the training loss, accuracy and validation loss.

However, they all resulted in the same validation accuracy 0.9411 which is the same as the logistic regression.

```
: model_3.summary()

Model: "sequential_1"

Layer (type)            Output Shape          Param #
==========================================================
dense_3 (Dense)         (None, 15)            165

dense_4 (Dense)         (None, 15)            240

dense_5 (Dense)         (None, 15)            240

dense_6 (Dense)         (None, 15)            240

dense_7 (Dense)         (None, 15)            240

dense_8 (Dense)         (None, 15)            240

dense_9 (Dense)         (None, 1)             16

==========================================================
Total params: 1,381
Trainable params: 1,381
Non-trainable params: 0
```

Neural Network

After investigation, we noticed that the prediction from neural network is basically all nonclick while around 10 clicks which is slightly better than logistic regression but not much difference.

Speculation

1. Since we dropped a lot of ID variables, We don’t have enough variables to capture some click features.

   Here is an example of this assumption:

Let's assume there is a young female customer who has senior shopping level and lives in a first class city. And let's see her behavior toward 5 similar products.

ad_position_type	cms_group_id	gender	age_level	shopping_level	occupation	new_user_class_level	price	btag	cate	click
0	11	2	3	3	0	1	2000	0	7971	0
0	11	2	3	3	0	1	1900	0	7971	0
0	11	2	3	3	0	1	2100	0	7971	0
0	11	2	3	3	0	1	1800	0	7971	0
0	11	2	3	3	0	1	2000	0	7971	1

But if we can know the information of the brand...

ad_position_type	cms_group_id	gender	age_level	shopping_level	occupation	new_user_class_level	price	btag	cate	click	Brand
0	11	2	2	3	0	1	￥2000	browsing	Shoe	0	Nike
0	11	2	2	3	0	1	￥1900	browsing	Shoe	0	Nike
0	11	2	2	3	0	1	￥2100	browsing	Shoe	0	Nike
0	11	2	2	3	0	1	￥1800	browsing	Shoe	0	Nike
0	11	2	2	3	0	1	￥2000	browsing	Shoe	1	Gucci

However...

There are too many unique brand ID.(around 100,000)

If we need to include the brand ID, we need to increase our data size as well.

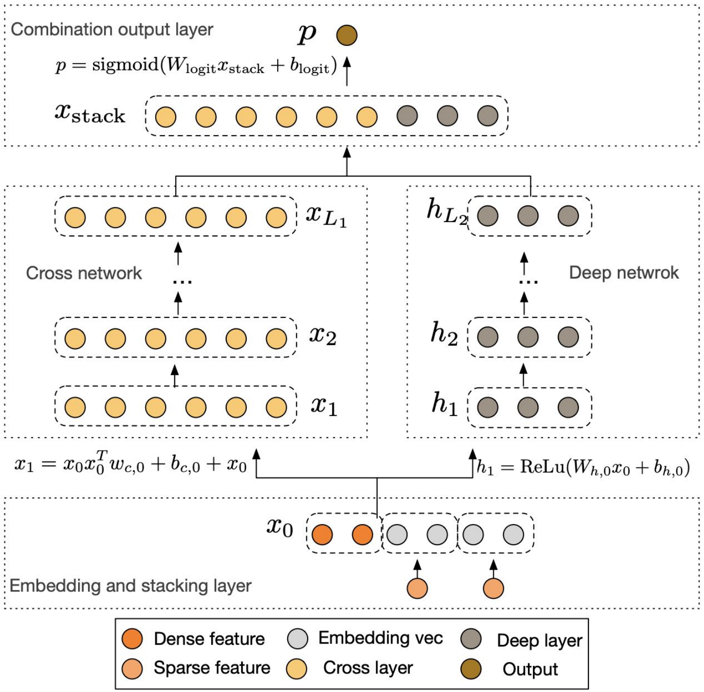
Our devices don’t have these high computational power.

Speculation

2. We need to apply more advanced neural networks such as Deep Interest Network (DIN), Deep & Cross Network (DCN), etc. which can greatly improve the expressive ability of the model and capture the diversity characteristic of user interests.

However, due to the computation power and the complexity of these models, we have troubles implementing these methods. Therefore, we decided to move to another method and see if we can improve the performance.

Deep & Cross Network



Support Vector Machine (SVM)

SVM

- Procedure
- Numerical result
- Visualization result

SVM - Procedure

- Y = click, X = features
- Training sets(75%), Testing sets(25%)
- Parameter gamma=0.1, optimal C=10
- SVM model
- PCA dimensionality reduction
- Result

SVM - Numerical Result

Confusion Matrix

Predicted Values

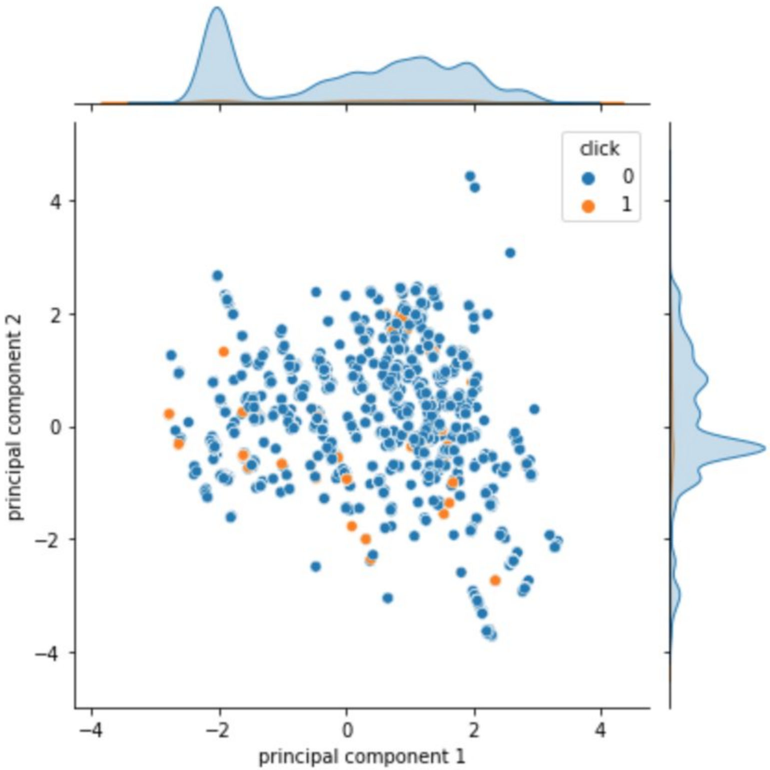
	Test set N = 15000	Negative 0 unclick	Positive 1 click
Actual Values	Negative 0 unclick	TN 14006	FP 111
	Positive 1 click	FN 529	TP 354

SVM - Numerical Result

Classification Report

	precision	recall	f1-score	support
0	0.96	0.99	0.98	14117
1	0.76	0.40	0.53	883
accuracy			0.96	15000
macro avg	0.86	0.70	0.75	15000
weighted avg	0.95	0.96	0.95	15000

SVM - Visualization Result



Conclusion and Discussion

Results and Future Studies

- The data we used is valuable
- The result using SVM has a better accuracy and faster performance than neural networks and logistic regression
  - Data volume
- 26 million data point
  - Requires better computational device
- Due to the limit of our computational devices, we were not able to perform Deep Interest Evolution Network (DIEN)
  - Requires more advanced device with larger graphics card

Thank you!