Data mining on kddcup2014

Shang Hang

Short discription:

for the data mining part, we use data from projects.csv, donations.csv, resources.csv,essays.csv. Our model is mainly based on the features extract from the projects and calculate the history variable based on the statistic. Then we use decision tree and the randomforest to do feature selection based on their feature importance,after that we tried randomforest classifer and gradiant boosting method to calculate the probability of the is\_exciting for the test.

Methodologies

data cleaning: we first read all the csv files including: projects.csv, donations.csv, resources.csv,essays.csv . After that we merge the projects and outcomes together for later use to calculate history features.

For the resources :

we aggregate the total cost of projects group by their projectId, item quantity of the projects, and also extract the average cost of the items required by the projects. After that, we also add one more attribute, price\_per\_student, which calculate the average money that was spend on the students reached by the project.

Then according to the resources type, we split the resources data into different trunks, and group by the project id on them. And calculate the sum of each type of resources need for the project. Since the different types resources’ cost will vary very large, like ipod and books.

For the essay:

I extract the length of the essay and length of the title for features

For the donations:

First I calculate the sum of donations total and is the teacher himself is also donor for each projects. Then we think of it might be helpful to get the average donations the projects get and the possibility of teacher being donor as well. So I calculate the history variable for each teach\_acctid and schooled, their average donation totals and average is\_teacher\_acct for later use.

For the projects:

First, I fill the NaN with padding , then I merged the projects with outcomes.

After merging, I transform the columns from the outcomes from ‘t’ and ‘f’ to 1 and 0, so that we could calculate.

For the features ‘three or more non teacher referred donor’,’one\_non\_teacher\_referred\_donor\_giving\_100plus’,

‘great\_messages\_porportion’, ‘teacher\_referred\_count’,’non\_teacher\_referred\_count’

I use them to calculate the features required by is\_exciting.

And for the one\_or\_more requirement , I added a new feature which is a summation of the three to see if this is greater than one or not.

I also add timestamp for each entry, split the date\_posted into year, month,day and also the days between and month between features which is calculated by subtract the reference date ‘2010-01-01‘

For the level features like grade\_level and poverty\_level, I manually change the level to 0 to 3 based on the string category

After these, I added tags for each entry indicating whether they are train, val or test.

To get the history features, I use several function to calculate the possibility of some important features that might possibly affect the probability of being exciting and other five primary requirements.

The idea is that I think a teacher which have lots of projects in the past and a high chance of those projects being exciting, he’s projects might be exciting in the future, also for the school, and the location where the school is, these features are probably important to indicate whether is exciting or satisfy these five requirements.

So I sum the data group by the ids, and school locations features, then count the occurrence. After summation, I calculate the possibility of outcome features occurrence given specific teacher or school, And I adjust the possibility by smoothing it with the mean. For those entries only occur once, assign the prob as 0.

Feature Selection:

After all the steps above, I save those result into separate files.

Read from the preprocessing result, I merged them together for training and testing

I first find all the categorical columns in the dataframe, and use LabelEncoder to encode those columns into integer.

Then started to fit the training data into the models.

I tried two different models for seeing feature importance, One is RandomForest, another one is DecisionTree. After fit the data into the model, I print out the sorted

Feature importance of them, then from the two result I got from the two model, I take the union of the two top 50 features, and use it as the feature might be important to the result.

Classification:

I have tried several methods to classify the test data, and seems that the RandomForestClassifier have the best performance of reaching accuracy of 0.604,

For the gradient boosting method, it takes a lot of time to run and the performance is not quite good as RF, only about 0.57 accuracy and might takes 2 hours to finished training while RF is done in few minutes.

I also tried AdaBoosting, logistic regression , naïve bayce classification, but none of these have better score than the randomforest. So I choose the randomforest as our final model and use it to compute the result

About tuning the params, We found that the val set was not quite useful, because it always have really high accuracy bigger than 0.9, so it’s hard to decide the param based on that. So we tried many times change the parameter of the RF classifier to get a better result.