COMP 4332 Project 1

Predicting Excitement at DonorsChoose.org

Lo Tsz Hin (20111661),

Ng Kam Pang (20122165)

Thusitha Chandrapala (20064923)

1) Introduction

DonorsChoose.org is an online platform where teachers could project ideas that require funding from external parties. The main task of this project was to predict exceptional ideas that had the potential of obtaining funding. A set of historical records were available as training data.

The following sections will outline how the raw data was processed, feature selection methods and learning algorithms used to obtain the final prediction.

The group tried to work on two solutions independently. In **Section 1**, we highlight the initial attempt based on the tutorial code provided. In **Section 2** we built another solution separately in a modular structure. The second attempt was able to obtain a better result.

kaggle.com registration:

- Team name : darthTC,
- user name: Thusitha Chandrapala

2) Section 1

At the start of the project, with many data sets and attributes, it is hard to start with right attributes and right method to make predictions. We selected few attributes as possible and a simple method to start the project.

a) Data cleaning

Data cleaning is very difficult in this project as there are almost missing values for each attribute. To fill in the missing values, we need to decide which method to be used. In the project, there are many kinds of attributes so we cannot use one simple method e.g. mean for numeric attributes. It is better to use different methods to fill in the missing values for each attribute. However, it is costly to determine the method for nearly fifty attributes and it is hard to measure the accuracy of the missing values filled. Thus, we use one method to fill the missing data for simplicity: previous data in the beginning of the project. Using this method, the data may be not very accurate because there may be no correlation between the previous data and the missing data so it is quite dangerous to fill it this way. But it is good for timely and simplicity reasons.

They are still many noisy data in the data sets, e.g. most of the data are the same for each attribute). We ignore these attributes as we want to observe the result first.

b) Feature selection

- Attributes used in Projects:
    - [primary_focus_subject, primary_focus_area, secondary_focus_subject, secondary_focus_area, total_price_excluding_optional_support, total_price_including_optional_support]
- Attribuets used in resources:
    - [project_resource_type]

We choose such few attributes because using all the attributes in the data learning process may adversely affect the result. The result may be bad because of the noisy data contained in the attributes and it needs so much time to make prediction using all the data in the data learning process. Thus, we choose a few attributes that can distinguish two different projects by observing the data (whether most of the data for that attributes are the same) and understanding the impact of the attributes (is it significant to distinguish the project).

Besides, we do not use neither all the data sets nor only one data set because it is neither time-consuming, useless data nor incomplete coverage of the data. It is better to select some of the data sets that are significant (variety of the project).

Although we use our own way to select the features, we cannot guarantee that the features are perfectly selected as there are so many combinations of feature selected. At the start of the project, we will make a try and decide the impact of these attributes.

c) Learning method

We use logistic regression to make prediction of the data (whether the project is exciting) which clearly is a Boolean type answer so it is simpler and more reasonable to use logistic regression rather than the linear regression or Bayes' rule.

d) Result

At the end of our first try, we have a result: 0.51 which is bad as it is almost like guessing the result randomly. From the result, we could say that the attributes selected do not determine whether the project is exciting. Then, we keep adding attributes into the prediction process and the score gets higher and higher. After adding almost all the attributes of the projects.csv, the score is around 0.56, which is

much better than our first try. And we realize that most of the attributes contribute to the final prediction of the project.

Besides, the score of only using logistic regression based on the tutorial code seem to reach a peak at around 0.56 so we decide to change our approach.

3) Section 2:

a) Obtaining and cleaning data

The Kddcup website provides a mixture of relevant and irrelevant data. The summery of the available data as given in the website is quoted below.

- **donations.csv** - contains information about the donations to each project. This is only provided for projects in the training set.
- **essays.csv** - contains project text posted by the teachers. This is provided for both the training and test set.
- **projects.csv** - contains information about each project. This is provided for both the training and test set.
- **resources.csv** - contains information about the resources requested for each project. This is provided for both the training and test set.
- **outcomes.csv** - contains information about the outcomes of projects in the training set.

Since using the data from 'projects.csv' did not give good results in Section 1, we decided to use the data from 'esseys.csv' as well. The data available in the resources.csv and donations.csv did not appear to have any useful features.

'Pandas' library is used for the major data processing activities. The summery of the data processing methods used are as below.

1. Data from the 'projects.csv' and 'essays.csv' are loaded into Pandas dataframes
2. The indexes based on the project IDs are obtained for the training and testing data
3. For the data in projects.csv file (both training and testing data), there are many missing values. For categorical data, the missing values are filled with the most common entry in the training data, and for numerical data, missing values are filled with the mean across samples of the training data.

b) Feature selection

From the total number of features available in the 'projects.csv' file, a certain subset was selected based on visual evaluation and test results.

Following features were discarded based on visual examination.

- secondary_focus_subject', 'secondary_focus_area' : There are a large number of missing data points for these fields. Therefore they would become problematic as features.
- 'Schoolid', 'school_ncesid': Too many unique values. Would result in over-fitting

Following features were discarded based on the test results

- 'school_latitude','school_longitude': The 'school_zip' would contain adequate geographical information. Since latitude and longitude are
- 'school_state': same argument as above
- 'teacher_acctid': contrary to the initial belief that this is an important feature, because successful teachers might have tendency to get more funding, the inclusion/removal of this feature did little difference.

The length of the essay was added as an additional feature, which had a positive impact.

c) Feature extraction from essays

The 'essays.csv' file contains descriptions about submissions in text format. There are three sub-fields we could use: Title, short description and an essay. Information was extracted using 'term frequency–inverse document frequency' method. The in-build module available in sklearn was used for this task. Surprisingly only using the essay information gave the best result, over using the essay and the short description.

d) Learning algorithms used

In this study, several machine learning algorithms were tested.

- Logistic Regression
- Random Forest
- Ada-Boost

- Gradient Boosting

Apart from Logistic Regression, all other methods used could be classified as meta-classifiers. These meta-classifiers are made up many simple classifiers (e.g. Large number of decision trees with very few branches)

Method	Score on Private leaderboard
Logistic Regression	**0.55950**
Ada-Boost	**0.58859**
Random Forest	**0.58222**
Gradient-Boost algorithm	**0.59176**

As a generic rule, we can see that meta-classifiers are much ahead of simple linear models like Logistic Regression. Therefore we use the best-performing option, i.e. Gradient Boost algorithm as the final method.

e) Combining essay features

Learning was carried out step by step using different models and feature sets. The available information was combined in different ways to obtain the final solution with the highest score. A summary of the results are given in the table below.

Method/data used	Score : public leaderboard	Comments
Logistic regression on features from 'projects.csv'	**0.55950**	
Logistic regression on features from 'essays.csv'	0.56080	
Gradient Boosting on concatenated features from 'essays.csv' and 'projects.csv'	•	The solution took too long to compute.
Logistic regression on concatenated features from 'essays.csv' and 'projects.csv'	**0.51848**	Worse performance than using individual features
Linear combination of Gradient Boosting on features from 'projects.csv' and Logistic regression on features from 'essays.csv'– Method 1	**0.58960**	Gradient-Boost algorithm alone performed slightly better . So still not a good way to combine features.
Gradient Boosting on combined features from 'essays.csv' and 'projects.csv' – Method 2	**0.59473**	LR on essays, and include the probability as a feature for the main classification. Performance

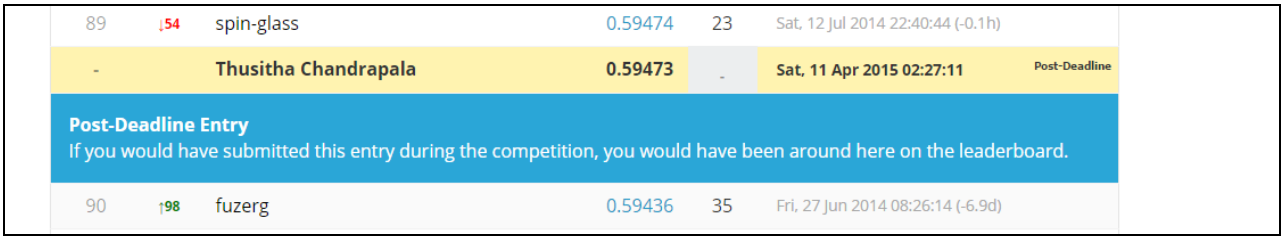
		Improvement! Ranked #90
Gradient Boosting on combined features from 'essays.csv' , 'projects.csv' with other info – Method 3	**0.59685**	LR on essays and other info, include the probabilities as a feature for the main classification. Performance Improvement! Ranked #81
Two level learning, Gradient Boosting and Logistic regression- Method 4	**0.60106**	Run GB on the resulting probabilities from previous predictions

i) Method 1

In this method, the essay features from tf-idf are used to create a Logistic Regression (LR) model. The features from 'Projects.csv' are used to make a separate model using a Gradient Boosting. The final result is a linear combination of the prediction probabilities (equal weights). The results from this method was not conclusive, as it was lower than the situation where only 'Projects.csv' features were used.

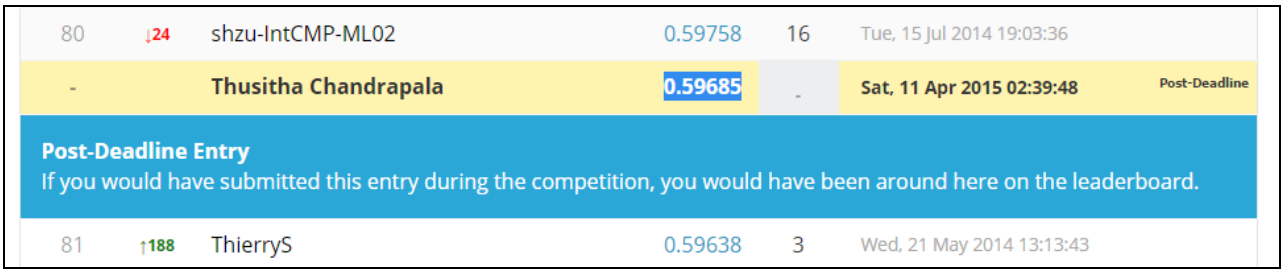
ii) Method 2

In this method, the essay features from tf-idf are used to create a Logistic Regression (LR) model. The estimation probabilities from the LR model are used as feature and concatenated to the set of feature vectors obtained from processing the 'projects.csv' file. A meta-classification method Gradient boosting is used for the main model to decrease variance. An improvement over using only 'projects.csv' is seen. Ranked #90 in the Private Leaderboard



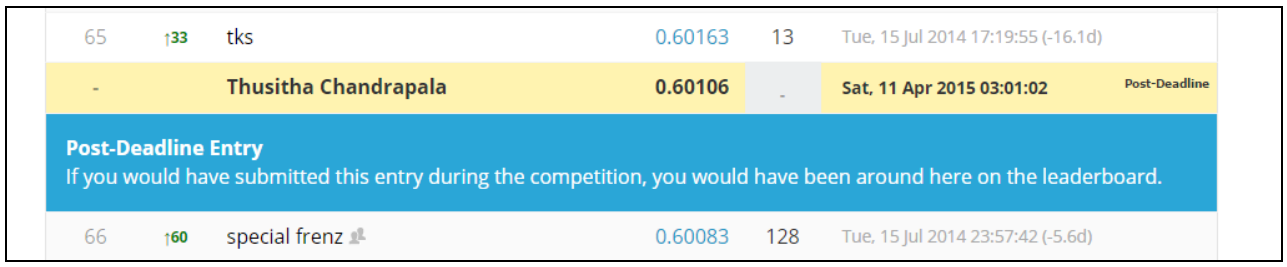
iii) Method 3

Similar to Method 1, but the binary fields in the 'outcomes.csv' are also used similar to the way essay features are used. Performance increase seen. Ranked #81 in the Private Leaderboard



iv) Method 4

This method uses a two level learning strategy. In the first level, a meta classifier is used to predict the class probability of being exciting from the features in 'projects.csv', and a separate LR model is used to predict the class probability of being exciting from the essay. Furthermore other outcomes from the 'outcomes.csv' file are also modeled using LR from the features given in 'projects.csv'. In the 2nd layer, the resulting probabilities are fed into another meta classifier which is used to predict if the project is exciting or not. This method gave the best results. Ranked #66 in the Private Leaderboard. It should be noted that the results are somewhat stochastic, and the scores change slightly on each run.



f) Parameter tuning for the Gradient boost algorithm

Slight parameter tuning proved to be useful in increasing performance. For the 1st level GB model, we found best results when the learning rate was set to 0.2 and the sub-sampling parameter was set of 0.4. The latter helps to offset the imbalance the data for the two classes. Inbuilt cross-validation methods as well as the final scores were used in the parameter tuning.

g) Non-successful attempts

It was tested weather including additional features like teacher history (i.e, Teacher has a successful 'is_exciting' entry before), and the month of the entry. But they did not give performance increases. Furthermore it was testes if one hot encoding of certain features would help increase the performance. With Gradient Boosting based learning this did not help much.

4) Code Explanation

Two individual codes are submitted.

- Codes for Section 1: test_code_1.py
- Codes for Section 2: The code consists of 3 main sections
    - Explore_seperate.py: The main function and handling code. This is the file to be run. Important functions are '_main_()' and 'readProcessData'.
    - dataProcess.py: contains the class "DataReader" which handles loading, cleaning, feature selection ... etc

- learnModel.py: contains the class “MlModel” which creates the learning models and carries out the training/prediction tasks

5) Online help and references

Some sections of the essay processing (Data cleaning part) was referenced from online resources (http://beatingthebenchmark.blogspot.de/).
Furthermore the KDDcup online forum provided valuable insights into the project.

6) Work Breakdown

The codes and the report for Section 1 were carried out by Lo Tsz Hin (20111661), Ng Kam Pang (20122165). The codes and report of Section 2 was carried out by Thusitha Chandrapala (20064923).

7) Conclusion

Even though there were code samples provided by winning contestants, we decided to approach the problem incrementally without copying and editing those entries, so that the learning outcome would be maximum. The initial attempt outlined in Section 1 could give reasonable results, but was not expandable. Therefore we decided to write a modular code with a separate data processing class and a machine learning class which would be reusable in other data mining projects as well. With this code, we built a hierarchical learning model and combined the features from essays.csv and projects.csv to obtain the final result.