Machine Learning Engineer Nanodegree

Kickstarter Success Prediction

Sujit Horakeri

February 22, 2018

Domain Background

Kickstarter is a community of more than 10 million people comprising of creative, tech enthusiasts who help in bringing creative project to life. Till now, more than $3 billion dollars have been contributed by the members in fueling creative projects. The projects can be literally anything – a device, a game, an app, a film etc.

Kickstarter works on all or nothing basis i.e. if a project doesn't meet its goal, the project owner gets nothing. For example: if a project's goal is $500. Even if it gets funded till $499, the project won't be a success.

I'm trying to build a binary classification model here to find if a project will reach it's intended goal of fund amount or not.

From the sited articles below, it is clear that to build a good classification model, we would need to consider variety of features like name, description, goal of the project, duration etc. Also, algorithms like Random Forest seem to have the best accuracy among others.

Related Works:

https://cseweb.ucsd.edu/classes/wi17/cse258-a/reports/a108.pdf

https://www.stat.berkeley.edu/~aldous/157/Old_Projects/Haochen_Zhou.pdf

https://www.kaggle.com/codename007/funding-successful-projects

Problem Statement

This is a binary classification problem. The aim here is to create a prediction model to predict whether the project will be funded or not i.e., we need to determine if the kickstarter project will be a success or if it won't be a success.

The inputs here are:

'project_id', 'name', 'desc', 'goal', 'keywords', 'disable_communication', 'country', 'currency', 'deadline', 'state_changed_at', 'created_at', 'launched_at'

The output should be:

'final_status' – 0 //failure

1 //success

The variable **backers_count** is only available in the training data and is not present in the testing data and hence cannot be used for training the model.

Datasets and Inputs

There are two files: train.csv, test.csv. The train data consists of sample projects from the May 2009 to May 2015. The test data consists of projects from June 2015 to March 2017.

Shape and memory info:

	Train.csv	**Test.csv**
Shape	108129, 14	63465, 12
Memory Usage	10.8+ MB	5.4+ MB

Dataset Details:

project_id: unique id of project

name: name of the project

desc: description of project

goal: the goal (amount) required for the project

keywords: keywords which describe project

disable communication: whether the project authors has disabled communication option with people donating to the project

country: country of project author

currency: currency in which goal (amount) is required

deadline: till this date the goal must be achieved (in unix timeformat)

state_changed_at: at this time the project status changed. Status could be successful, failed, suspended, cancelled etc. (in unix timeformat)

created_at: at this time the project was posted on the website(in unix timeformat)

launched_at: at this time the project went live on the website(in unix timeformat)

backers_count: no. of people who backed the project

final_status: whether the project got successfully funded (target variable – 1, 0)

There should not be any missing data issues as the dataset contains very less number of null/nan values. There are 3 null items in the **name** columns of **training** data and 9 missing values in the **desc** column. While the **test** data has 4 missing **desc** entries

There is an imbalance in the target variable distribution as there are more entries with unsuccessful values than successful. Below are the value counts

Unsuccessful (0) 73568

Successful (1) 34561

The data is gotten from:

https://www.kaggle.com/codename007/funding-successful-projects/data

https://www.hackerearth.com/problem/machine-learning/funding-successful-projects/description/

Solution Statement

I am planning to use the ensemble algorithms like GBM, RandomForest for this classification problem as I feel that this could involve a lot of features since we have to deal with a variety of categorical data along with text data. We would also need to use text processing techniques to determine how the text features - name, keywords and description affect the final outcome of the project getting funded or not. We could use techniques like tokenizing, stemming and lemmatization for processing of text data. Another option is to use pretrained models like Word2Vec for the text.

Benchmark Model

Since this is a publicly available dataset there isn't a benchmark model as such, but we could use RandomForest as a benchmark model. I will then try to beat the performance of this RF model with other ensemble algorithms like the GBM.

We could compare the RF model and the GBM model to find out how they both compare. Stacking of multiple algorithms can be performed or even same algorithm with different hyper parameters can be used as well to make use of each models' strengths and produce a more accurate final output.

Evaluation Metrics

The evaluation metric proposed in the competition is **accuracy score**, I'll be using accuracy score as the evaluation metric.

Accuracy is defined as the percentage of the predictions that were accurate.

Accuracy = (true positives + true negatives) / total population

We could also use other metrics like F score in combination with Accuracy score as accuracy score alone might be misleading sometimes. And since we have an imbalance in the distribution of the target variable in the dataset at hand, it is better to consider both Accuracy and F1 score as the evaluation metric.

F1 score is the harmonic mean of precision and recall:

2*precision*recall / (precision + recall)

Where,

Precision = True Positives/ (True Positives + false positives)

Recall = True Positives/ (True Positives + false negatives)

Project Design

I would proceed with the project in the following manner:

1. Data Exploration
2. Data Preprocessing: Handle missing data and treat outliers
3. **EDA** to understand the data better and do **feature engineering**
    - **Engineer new features from the text fields (name, desc, keywords) like the word counts in each text field.**
    - **Check for cancelled projects based on the deadline and status changed dates**
    - **Duration of the project**
    - **Project launch month, day of week and day of month**
    - **Use count/tfidf vectorizer to generate most frequently used words**
4. Split data into test and train
5. Evaluate base model[1] performance

6. Train a more robust model like RandomForest as this is our benchmark model
7. Train other models like GBM and compare with the benchmark model
8. Tune parameters.
9. Compare final model (GBM model) with benchmark model.

[1]Since this is a binary classification problem, we could start with predicting all success or all failure and check the accuracy, this would be the baseline model.