Proceedings of the Third International Conference on Computing Methodologies and Communication (ICCMC 2019)
IEEE Xplore Part Number: CFP19K25-ART; ISBN: 978-1-5386-7808-4

Success Prediction using Random Forest, CatBoost, XGBoost and AdaBoost for Kickstarter Campaigns

Siddharth Jhaveri[1], Ishan Khedkar[2], Yash Kantharia[3], Shree Jaswal[4]
Department of Information Technology, St. Francis Institute of Technology, Mumbai, India
[1]siddujhaveri@gmail.com, [2]ikhedkar12@gmail.com, [3]yashkantharia17@gmail.com, [4]shreejaswal@sfitengg.org

***Abstract*— Kickstarter is a popular crowd-funding platform used by individuals and groups to obtain the seed amount to implement their business ideas. It is essential for the campaign owners to plan their crowd-funding campaign beforehand in order to avoid failure. This paper aims to predict the success or failure of a Kickstarter campaign and help the campaign owners plan better for their campaign by providing trends and analysis based on historical data of campaigns on Kickstarter, ranging from 2014 to February 2019. Various classification and boosting algorithms were applied and it was concluded that Weighted Random Forest along with AdaBoost for the subsampled dataset gives the best accuracy.**

Keywords*— *Crowdfunding, Kickstarter, Success Prediction, Classification.

I. INTRODUCTION

The past decade has witnessed the start-up revolution. Due to the increase in technical advancement and accessibility to knowledge resources, individuals possess the potential to generate products to fulfil ever increasing market needs. The only hurdle they face is initial capital to back their idea. Crowdfunding is a way of raising capital through public support. This helps individuals and groups to gather support and spread awareness about their business ideas and let them begin realization of their products. Generating the seed amount from crowdfunding empowers and encourages budding entrepreneurs who lack financial backing. Crowdfunding also works as a tool to identify market demands and customer's interest related to your products, services or ideas.

Usually, crowdfunding has various types: reward-based, donation-based, equity-based and lending-based. Reward-based projects are the most common and offer the funders value in terms of some product or service under their campaign directly based on the amount the funder chooses to raise for the campaign. Donation-based campaigns do not provide any product or service to the funder but credit and recognise them for the donation they pledge through gestures and mentions across their ecosystem. Equity-based projects directly equates the amount a funder pledges with the shares of the company/stakes in the project at hand. Lending-based projects serve as a temporary amount used by the project owners to use for their work or provide backing for the demand from the customers; but the funders can claim their money back after a default period, with either rewards or compensation as interest paid by the campaign company [1][2]. Also, based on the end-goal and type of fund summation, there are two types: All-Or-Nothing and Keep it All. All-Or-Nothing assures that the campaign starters only receive the pledged amount if they reach their goal. On the other hand, Keep-It-All allows the campaign starters to keep the amount raised up till the deadline even if their goal is reached and the campaign is termed as 'failed'[3].

Kickstarter provides a crowdfunding platform to all these creative minds. These creators can start a campaign by setting project's funding goal and deadline. If people are interested about a project they can support it by pledging a certain amount of money towards the campaign. When the campaign goal is reached, the amount is collected from the backers and fee is charged from the project creator. If the campaign fails, no one is charged. The success is usually influenced by the initial factors set for the campaign such as deadline, amount goal, category of the project etc.

II. RESEARCH REVIEW

Hussain, Kamel, & Radhakrishna [4] tried to predict the success rate of a campaign from their dataset consisting of around 40,000 successful project's information out of a total of 130,000 campaigns in the dataset, since only about 40% projects attain success. They implemented various types of algorithms: K-nearest, Logistic regression, Random Forest and Support Vector Classifier. Their research paper consisted a combination of simple and complex features used to divide various parameters: Time, Location, Category, Text, Goals, Number of Goals/Rewards and Staff picked, etc. Their paper concluded that Random Forest was the best approach to analyse the dataset with an accuracy of 80.67 considering all the complex features too. They suggested to implement any boosting method to the obtained results to further improve the accuracy.

Mollick, Ethan [5] explained the basic understanding of how Crowdfunding works in the first place, it's use in the current ecosystem and the various goals of the funders and the founders of various campaigns. The paper highlights the need to appropriately target the goal for the campaign, because if a campaign is successful, it succeeds by a small margin, but if it fails, the margin is substantially higher.

Felipe, Mendes-Da-Silva, Gattaz [6] present a study similar to the one seen in [3] with respect to the roles mass media and geography play as the success factor for any Kickstarter campaign. It helped to understand the probability the founders can assess for financial returns based on their media activity and response rate and the role location plays of the project's initiation and place of work.

Li, Rakesh & Reddy [7] predicted the project success, using newer prediction models that combine the power of both classification (which incorporated both successful and failed projects) and regression (for estimating the time for success). Their notable feature was to use censored regression techniques to inculcate successful as well as failed projects as opposed to other regression methods that only work on successful projects, thus increasing their accuracy. Their feature list includes the number of projects created, projects backed by the creator, success ratio of the creator, and 5 features obtained from creator's Facebook profile, the duration of project, the goal amount, the number of images,

the presence of videos and the number of comments about the project. They used Cox proportional hazards model, Tobit regression, Buckley-James estimation & Boosting concordance index. They concluded saying that Log-logistic method is the more accurate approach and also using censored regression gives far superior results. Also they stated that the status of a project in the first 3 days during its funding stage really matter its total run.

Ahmed, Tyagi, Kaur [8] used a dataset of 26,000 projects and applied various classifiers to predict the success rate and conclude that Random Forest is the best for the job at hand based on their experimental values. They used an approach wherein they used weight training to optimize their forest after each run. This helped us understand the importance and improvement in results after applying weights.

III. DATASET

A. *Features Used*

The dataset was compiled using the data provided by Webrobots.io [9], a web scraping company that has datasets of Kickstarter projects from 2014 till February 2019. It consists of features like: Name, ID, Blurb, Launched at, Finished at, Main category, Sub-category, location, FX rate, currency, amount pledged and raised, etc. But the data consisted of dictionaries as input data in certain features that had to be cleaned and optimized as new features in order to make this data more usable and easy to compute. The final Dataset consists of the following features:

- Blurb Length: How many words are used to explain the project in short
- Duration: Specify the duration of the campaign in days using the date of launch and date of completion
- Main category: Define what category the project represents to better highlight its uniqueness
- Sub-Category: Sub-category within the main category.
- Currency: The currency the project holder requested its backers to fund in
- Location: Country, State and City of origin of the campaign
- Goal: The goal defined by the project holder to be fulfilled before the duration ends in USD.
- Name Length: Using the number of words in the name of the project

The above mentioned features will serve as input features for our Machine Learning model, upon which the pre-processing will be applied to fit the data as per the computation needs.
A dataset has been also published by us on Kaggle [10] so that anyone can refer to our dataset and use it for their own research.

B. *Data Analysis*

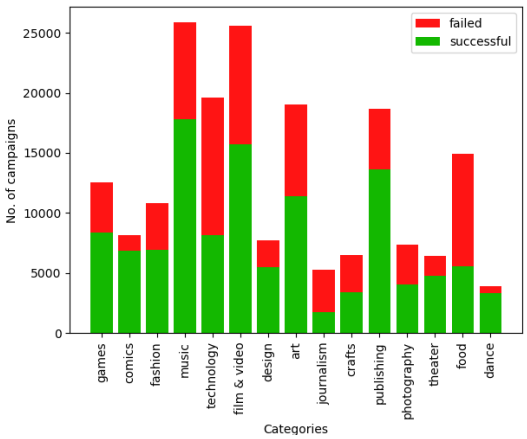

Figure 1: Distribution of campaigns by category

The analysis gives us a fair insight as to how the values are distributed across the range and highlight skewed data and any irregularities if present.
The amount of failed and successful campaigns with respect to each Main Category is illustrated in Figure 1 to understand that our dataset is a bit skewed towards successful campaigns.

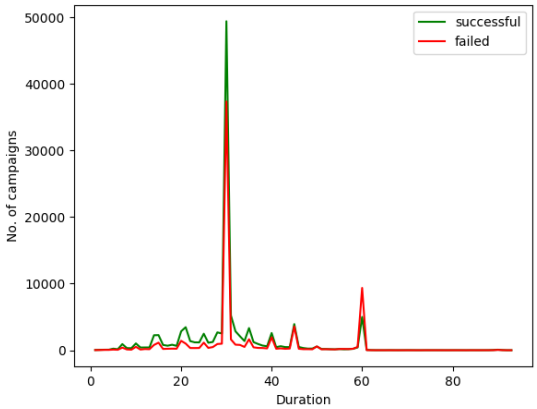

Figure 2: Distribution of campaigns by duration

In Figure 2 we see a trend that usually campaigns with duration between 10 to 40 days have higher success rates and after 40 days there is decline in success rates.

IV. METHODOLOGY

A. *Algorithms referred*

The following models were used for classification: Random Forest, XGBoost and CatBoost, also AdaBoost was applied on the Random Forest model. Parameter tuning for each of them was done to get the optimal output.

1. Random Forest (RF)

It is a supervised learning algorithm which works by creating a forest of decision trees and merges them to form a more accurate model which can be used for both classification and regression. [11]

-Weighted Random Forest (WRF)

The introduction of weights is to apply a cost sensitive learning model. The trees individually assign a weight to each class and heavily penalise the minority classes, where the

weights are used at root nodes for splits and at terminal nodes for assigning priority. At the end, the aggregate of weights from each tree is considered for the final prediction call. This will greatly help us reduce the effect of imbalanced data on our model [12].

2. AdaBoost

AdaBoost combines multiple weak models to generate one strong model. It does so by creating one weak model on the training data and then it creates models iteratively, where each model reduces the errors of its previous model, till maximum accuracy is obtained or defined number of models are added [13]. AdaBoost with the models generated by Random Forest as base estimator, produced the best result.

3. XGBoost

XGBoost uses gradient descent on decision trees to generate multiple models which are combined sequentially while correcting the previous models to generate a final optimal model. XGBoost is very efficient in terms of usage of computational resources and processing speed [14].

4. CatBoost

CatBoost is a gradient boosting library which is capable of handling categorical data. It does not use binary substitution of categorical values, instead it performs a random permutation of the dataset and calculates the average label value for the example with the same category value placed before the given one in the permutation. CatBoost reduces overfitting with this approach of handling categorical features [15].

B. Experimentation

The algorithms were applied on the dataset's features and the prediction along with the accuracy, precision and recall were calculated for each algorithm, for comparison. We used a 90:10 split ratio for our training and test set, respectively. Also, we chose a different approach than the previous work done; we subsampled our dataset into smaller sets based on the Main Category of the project and generated a model for each subsample. This approach was applied because this reduces the number of categorical features by 1 and improves the accuracy of individual models for a certain category. This results in 15 models, one for each category, as shown in Figure 3.

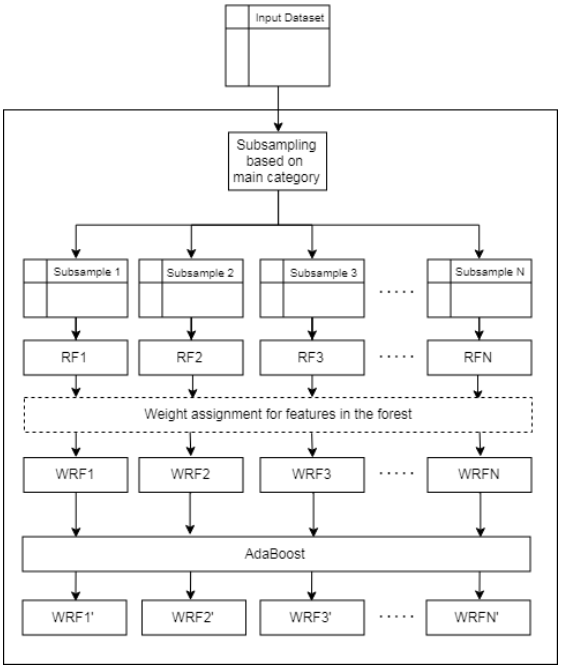

Figure 3: Generation of the models

For this approach it is essential to use the correct classifier based on the main category of input values as, shown in Figure 4.

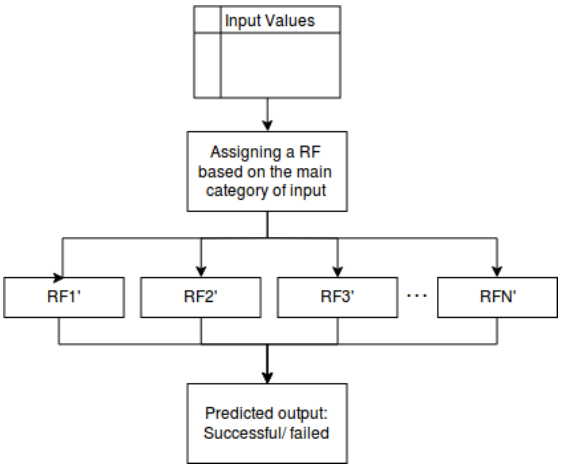

Figure 4: Predicting for input values

V. EXPERIMENTAL RESULTS

Table 1: Results without sub sampling.

Classification Model	Accuracy	Precision	Recall
Random Forest (γ=120)	79.97	0.80	0.64
Random Forest (γ=120) with AdaBoost (γ=20)	80.51	0.82	0.63
XGB Classifier (γ=10)	74.29	0.76	0.49
CatBoost (Learning rate = 0.21)	83.33	0.80	0.75

Table 2: Accuracy of models using subsampling.

Main Category	RF	RF + AdaBoost	WRF + AdaBoost
games	88.06	88.78	88.62
comics	87.63	88.00	87.88
fashion	80.03	80.22	80.31
music	78.87	78.91	79.38
technology	79.69	80.71	80.61
film & video	81.03	80.60	80.72
design	85.10	84.97	85.10
art	74.64	74.54	74.38
journalism	92.06	93.57	93.00
crafts	92.18	92.49	93.10
publishing	83.40	83.13	84.15
photography	85.57	86.39	87.21
theater	80.18	80.03	81.26
food	83.08	83.96	84.56
dance	90.07	88.80	91.60
Average	**84.10**	**84.34**	**84.79**

(No. of estimators for Random Forest =120, No. of estimators for AdaBoost =120, Weight for Random Forest = 150)

VI. Conclusion and Future Work

We experimented with the above-mentioned models. XGBoost executes quickly but suffers from overfitting due to binary substitution using one hot encoding. Hence, we used CatBoost which executed quickly with significantly better results than XGBoost. Random Forest when used with AdaBoost on the complete training data gives less accuracy, but when we subsample the dataset based on the Main Category and take average of the individual accuracies, we get an accuracy which is greater than the rest of the methods.

Overfitting due to binary substitution of categorical features, while using Random Forest, can be reduced by subsampling the training dataset based on common features and hence the accuracy is increased, but at the cost of computational time. Our research helps to conclude that the WRF with AdaBoost model gives the best average accuracy at 84.79%.

Future Work

The accuracy of any model depends on the training data. For Success Prediction we have considered only successful and unsuccessful projects, not the cancelled and suspended ones. There are many more possible outcomes and features that influence the probability of a project. Further studies can be made by:

- Using social-media posts and references published by the campaign before and during the runtime to study impact on the success rate of a campaign
- Understanding inflation in success rate w.r.t the advertising campaign of a project, through offline and online methods
- Including the Cancelled and Suspended category of campaigns for better understanding of the market.

References

[1] Bannerman, S. (2013, 6). “Crowdfunding Culture. Journal of Mobile Media”, 7(01), pp: 1-30. Retrieved from Wi: Journal of Mobile Media

[2] Giudici, G., Nava, R., Rossi Lamastra, C., & Verecondo, C. . (2012). “Crowdfunding: The new frontier for financing entrepreneurship?” Milano.

[3] Cumming, D., Leboeuf, G., & Schwienbacher, A. (2014).“Crowdfunding models: Keep-it-all vs. all-or-nothing.”

[4] N. Hussain, K. Kamel and A. Radhakrishna. ”Predicting the success of Kickstarter campaigns”, Cseweb.ucsd.edu, 2018. [Online].

[5] Mollick, Ethan. ”The dynamics of crowdfunding: An exploratory study.” Journal of business venturing 29.1(2014): 1-16.

[6] Felipe, Mendes-Da-Silva, Gattaz. “Crowdfunding Research Agenda”, 2017 IEEE 11th International Conference on Semantic Computing:459-464

[7] Li, Rakesh & Reddy. ”Project Success Prediction in Crowdfunding Environments”. Ninth ACM International Conference on Web Search and Data Mining,pp. 247-256, CA, USA. (c) (2016)ACM.

[8] Ahmed, Tyagi, Kaur, ”Predicting Crowdfunding Success with Optimally Weighted Random Forests”. IEEE - ICTUS (2017):769-775

[9] Webrobots . “Kickstarter Datasets” https://webrobots.io/kickstarter - datasets.

[10] Kaggle . “Kickstarter Campaigns” https://www.kaggle.com/yashkantharia/kickstarter-campaigns.

[11] Breiman, L. (2001). “Random forests”. Machine Learning, 45, 5–32.

[12] Chen, Liaw, Breiman. “Using Random Forest to Learn Imbalanced Data”. UCB-Report No: 666(2004)

[13] Scikit-learn.org . “sklearn.ensemble. AdaBoostClassifier — scikit-learn 0.20.3 documentation” https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.AdaBoostClassifier.html

[14] XGBoost . “XGBoost Documentation — xgboost 0.81 documentation” https://xgboost.readthedocs.io/en/latest/

[15] Dorogush, Ershov, Gulin. “CatBoost: gradient boosting with categorical features support”. Yandex.