Predicting Crowdfunding Success with Optimally Weighted Random Forests

Fahad Sarfaraz Ahmad[1], Devank Tyagi[2], Simran Kaur[3]

[1]*Electronics & Communication Engineering Delhi Technological University New Delhi, India fahad bt2k14@dtu.ac.in*

[2]*Dept. of Computer Engineering Delhi Technological University New Delhi, India ank@gmail.com*

[3]*Dept. of Computer Engineering Delhi Technological University New Delhi, India simrankaur1509@gmail.com*

Abstract: ***The social media version of fundraising, crowdfund- ing has, within a very short time, become an important and often critical element of the process that allows budding entrepreneurs to grow in todays market. While the underlying principle behind crowdfunding - to get multiple people to financially back an idea and collectively amass enough resources to make it happen - is simple enough in theory, it is much harder to actually implement. Kickstarter, the largest and most well-known crowdfunding website internationally, sees more than half of the uploaded projects rejected due to inadequate funds gathered. In this study, we take a closer look at the reasons for success and failure of these projects from the project description pages, where the creators attempt to persuade potential backers into supporting their respective creations. A dataset of over 26 thousand Kickstarter projects was created and used to understand the factors that most affect the chances of acquiring successful funding. In this study, we propose a novel algorithm for the RandomForests learning model in which we assign optimal weights to the individual classifiers and perform weighted majority voting on them using the 13 predictors that were found to be most suitable through implementation logic. These predictors will serve to be useful to both, a project creator as well as a potential backer, in order to devote their energy and resources judiciously. Finally, we tested our optimized learning model on the freshly acquired dataset and observed 94.289% accuracy, 94.5% precision, 94.3% recall and 94.3% F-measure value in success prediction. This research achieved the best accuracy and greatest amount of success in this field.***

Keywords: Crowdfunding, Kickstarter, learning models, Decision Trees, RandomForests

I. INTRODUCTION

Crowdfunding has been around as a useful concept since 2007, but was able to gain traction only after the advent of websites such as Kickstarter, which allowed everyone from experienced creators to novice experimenters to put up their ideas in front of a large group of people and allow them to decide the viability of the ideas. However, while experienced creators may know how to acquire the required funding for the fruitful completion of their projects, the novice creators usually struggle to become successful, even if their ideas are viable as well as productive.

Online crowdfunding websites such as Kickstarter work in an all-or-nothing manner - if the funds collected for the project in the stipulated time match or exceed the creator-defined goal, the project creator gets the funding and his/her project is successful. However, if the funds collected fall short of the goal, even by a dollar, the creator will get nothing and the project is declared to be a failure. This strategy means that novice creators usually end up on the wrong side of successful funding [9], since it is difficult to know what 'works' for the large pool of potential backers out over the World Wide Web. According to Kickstarter, as of July 2017, there have been 367,763 projects uploaded, out of which only 130,209 projects have received successful funding - a success rate of merely 35.83% excluding live projects.

TABLE I: Project Statistics On Kickstarter.

Total projects	367,763
Successful projects	130,209
Unsuccessful projects	233,207
Live projects	4,347
Total money pledged	$3.23B
Money pledged successfully	$2.84B
Money pledged unsuccessfully	$356M

Hence, an urgent need for a guideline is apparent, one that supports novices and experts alike in seeing whether their project description and presentation are conducive to receiving successful funding or not. This research is thus motivated to provide this guideline in the form of a machine learning tool, which identifies the individual traits of a crowdfunding project and shows them to the creator, giving him/her the time and opportunity to make revisions to their projects before

978-1-5386-0514-1/17/$31.00 ©2017 IEEE

launching them on the online crowdfunding platform. This tool is prepared and tested for efficiency using the very large amount of data that is present from past projects, both successful and unsuccessful.

II. DATASETS

The website *kicktraq.com* contains the URLs of all the project description pages that have been uploaded on Kickstarter's servers. These URLs were used to automate the process of web scraping the data on the project description pages. The final dataset created contained the details of 26,191 Kickstarter projects. All these projects had their creation dates lie from 01/12/2009 to 20/03/2017. The dataset contains details of projects over 15 categories, with 12,911 of these projects having proven successful. While this is in contrast to the actual success rate of projects on Kickstarter, a higher number of successful projects were required by the learning model since success is the driving force behind the model's functioning. The web scraping process is illustrated in Figure 1.

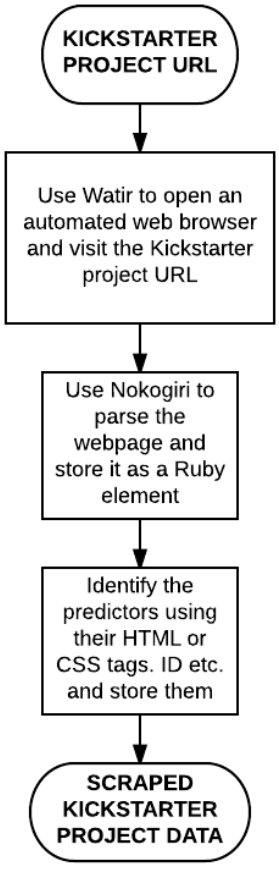

Fig. 1. The web scraping process to create the data

TABLE III: Learning Model Predictors.

Project Category	The field in which the idea/product falls
Goal	The creator-defined goal for accumulated funding received
SmogMain	The readability index of the project description given
SentMain	Number of sentences in the project description given
SentReward	Number of sentences in the reward description given
#Images	Number of images on project description page
#Videos	Number of videos on project description page
#Rewards	Number of rewards given by user
#Websites	Number of webpages related to the project
#Collaborators	Number of project collaborators
#FbFriends	Number of Facebook friends the creator has
#CreatedProjects	Number of projects created by project creator
#BackedProjects	Number of projects backed by project creator

TABLE II: Dataset Description Statistics.

Total projects scraped	26,191
Successful projects scraped	12,911
Unsuccessful projects scraped	13,280
Ratio of successful projects scraped	0.493
Creation time of first project scraped	01/12/2009
Creation time of last project scraped	20/03/2017
Number of predictors in dataset	13

The dataset consists of 13 predictors, attributes of data obtained from the project description page. Some of the predictors have been tested in past literature such as Greenberg [1] and Chung [2]. However, a large part of this research has been coming up with newer, better predictors that have hitherto been untapped for the purpose of learning. For example, one of our predictors is the SMOG Index of the project description [4] [6], which allows us to calculate the readability of a piece of text. Mitra and Gilbert [10] illustrate how the phrases used in the project description go a long way in determining the

success of a project due to its comprehensibility and tone. The 13 predictors are given in Table III.

III. EXPERIMENTAL DESIGN

A. Proposed Approach

The problem posed in front of us is twofold. Firstly, we must attempt to create a predictive model that can accurately classify the data points into the classes of success and failure. Secondly, we must find the factors that most affect the chances of success of the project, in order to allow the project creator to focus on those aspects of the project description page. Keeping these requirements in mind, the Decision Tree [11] learning algorithm was decided upon as a suitable solution. The Decision Tree algorithm is a simple inductive algorithm which is visualized in the form of a top-down tree, wherein each node except the root node is formed as a result of the parent node splitting, on the basis of some irregularity or disorder in the data. The leaf nodes of the Decision Tree are the results obtained, and each path in the Decision Tree corresponds to an independent decision-making logic. This ensures that not only do we get a class as an answer (in this case, the classification into success or failure), we also obtain the decision-making logic and the importance of each decision (in this case, the predictors most important in achieving the accuracy in prediction).

The accuracy obtained by the Decision Tree classifier, however, can be improved. Chandra et al. [14] and Soares et al. [15] discuss how ensembles of classifiers tend to give better results for classification accuracy than standalone classifiers.

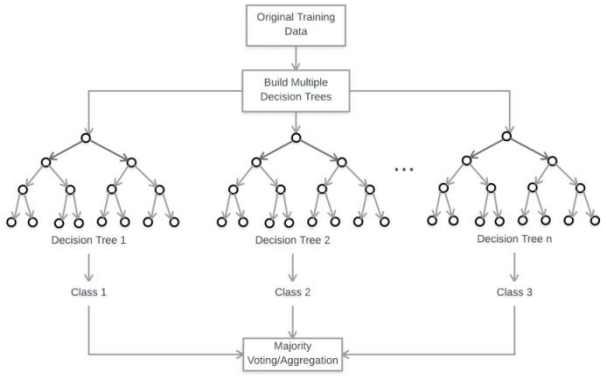

Fig. 2. The Conventional RandomForests Learning Model [5].

Hence, a randomly distributed ensemble of Decision Trees was decided upon as the learning model which would work upon our dataset. This is known as the RandomForests learning model [12], as shown in Figure 2.

The RandomForests learning model creates a number of independent Decision Trees with low correlation between the classifiers, with different combinations of predictors. These combinations of m predictors, where m<M (the total number of predictors to be used) is defined manually, are generated randomly by a meta-algorithm known as *random subspace method*. New training sets are generated by randomly sampling data from the original training set with replacement and using them to train the model. This is known as *bootstrap aggregating*, and it reduces the variance that Decision Trees, Neural Networks and other complex algorithms generally suffer from.

The RandomForests learning model evaluates the class of

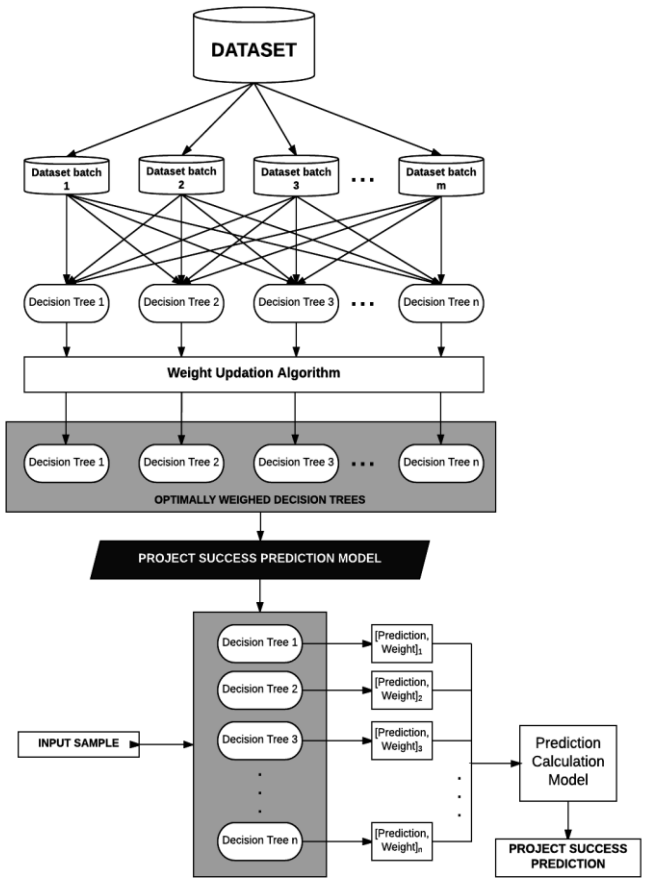

Fig. 3. Proposed Architecture of Optimally Weighted RandomForests Model, with *m* batches of data and *n* Decision Trees in the ensemble.

the data by taking the results of all the Decision Trees and conducting majority voting. It is easy to reason that some Decision Trees will return successful predictions with a higher accuracy than other trees, due to the random set of predictors bagged by them. It logically follows that the decisions taken by these trees should be considered with more importance than the decisions taken by trees with lower rates of accurate predictions. Winham et al. [7] and Lee et al. [8] used the out-of-bag (OOB) error rates to predict accuracy and assigned these accuracies as the weights of the Decision Trees. Our proposed model for the RandomForests learning model on this dataset uses a unique approach to assign greater weights to

the decisions made by high-accuracy Decision Trees in the ensemble.

Figure 3 shows the architecture of the proposed model. The Kickstarter projects training set is divided into m batches of projects, whcih are given batchwise to the conventional RandomForests learning model for training. Every Decision Tree performs the classification procedure and obtains an accuracy measure, A_c. This value of accuracy is compared against the accuracy of the ensemble, A_e to give a decision parameter, θ.

$$\theta = \frac{A_c}{A_e}$$

The decision parameter θ was used to decide the magnitude of weight updation required. If $\theta > 1$, the ensemble is working at a lower accuracy than the classifier itself, and that particular Decision Tree must be considered with a much greater weight than other trees. Similarly, if $\theta < 0.33$ (empirically determined value), the weight of this Decision Tree's predictions must be drastically reduced in order to increase the ensemble's overall accuracy. In this manner, all the trees are given updated weights on the basis of their individual accuracy rates. The decision parameter θ is scaled upto the weights of the trees by a weighing factor, β, which we take as the running average of the weights of trees till that point.

$$\theta = \theta * \beta$$

Once all the Decision Trees in the ensemble have been trained in this manner, the test set is passed to the ensemble learning model, where each Decision Tree classifies the data given into the class of either success or failure. These decisions, D_i, are multiplied by the weight of their respective Decision Tree to give the weighted vote, V_i. Weighted majority voting is then performed to decide the final class of the project.

$$V = \sum_{i=1}^{n} V_i$$

Algorithm 1: Weight Updation

Input: treeForest: ensemble of Decision Trees used for classification, the weight of each of which is initialized as 1, n: number of Decision Trees, m: number of batches of data from dataset

Output: treeForest: ensemble of Decision Trees with weights appropriately updated

1. obtain m batches of data from original dataset
2. $\beta \leftarrow 1$ (average weight of all Decision Trees)
3) for i ← 1 to m (for each batch) do:
4) numData ← number of project descriptions in batch
5) for j ← 1 to numData (for each project) do:
6) for k ← 1 to n (for each Decision Tree) do:
7. $\theta \leftarrow A_c/A_e$
8. $\theta_1 \leftarrow \theta * \beta$
9. if $\theta > 1$ weight[k] ← weight[k] + $(\theta_1 * n)$
10. else if $\theta > 0.66$ weight[k] ← weight[k] + θ_1
11. else if $\theta > 0.33$ weight[k] ← weight[k] - θ_1
12. else weight[k] ← weight[k] - $(\theta_1 * n)$
13. $\beta \leftarrow$ update running average with new weight
14) return treeForest with newly weighted Decision Trees

Algorithm 2: Prediction Calculation

Input: treeForest: ensemble of Decision Trees used for classification, with optimally weighted Decision Trees, n: number of Decision Trees, testSet: the test set of data on which accuracy is derived

Output: Success: binary variable denoting 1 for success and 0 for failure

1) numData ← number of projects in testSet
2) for i ← 1 to numData (for each project) do:
3) initialize zeroc and onec as 0
4) for j ← 1 to n (for each Decision Tree) do:
5) pass testSet[i] to treeForest[j]
6) c ← result of classification by treeForest[i]
7) if c is 0, zeroc ← zeroc + weight[j]
8) else onec ← onec + weight[j]
9) if zeroc > onec success ← 0
10) else success ← 1
11) output success

The web scraping and classification were performed on Intel Dual Core i7-7500U (upto 3.5GHz), with 12GB DDR4-2133 SDRAM and Ubuntu 16.04 operating system. The classification was performed over the dataset described above, with a 66-34 train-to-test split after resampling the data to avoid sampling bias.

The classification process was divided into two phases.

- The first phase involved the classification of the entire dataset. In this phase, we retained "Project Category" as a predictor.
- The second phase involved classification over 15 different datasets, partitioned on the basis of their project categories,to test whether projects of different categories could be predicted with more accuracy by different pre-

dictive models for each category separately. In this phase, "Project Category" was not retained as a predictor.

B. Evaluation Metrics

- ***Accuracy:*** Accuracy is one of the most popular evaluation metrics, used for predictive models in most literatures. In terms of the confusion matrix, accuracy is defined as the sum of the true positives and false negatives (number of correct predictions) over the total number of predictions.

$$\text{Accuracy} = \frac{tp + tn}{tp + tn + fp + fn}$$

- Precision: Precision is an evaluation metric that gives a measure of the number of relevant predictions from amongst the total predictions made. In terms of the confusion matrix, precision is defined as the number of true positive predictions over the total number of positive predictions made.

$$\text{Precision} = \frac{tp}{tp + fp}$$

- ***Recall:*** Recall is an evaluation metric that gives a measure of the number of relevant predictions made from amongst the actual relevant instances. In terms of the confusion matrix, recall is defined as the number of true positive predictions made over the total number of positive instances.

$$\text{Recall} = \frac{tp}{tp + fn}$$

- ***F-Measure:*** F-Measure or F-score is also one of the most popular evaluation metrics used for predictive models. The F-Measure is calculated by first calculating the precision and recall values of the model and then finding the harmonic mean of these values.

$$\text{F-Measure} = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall}$$

IV. EXPERIMENTAL RESULT ANALYSIS

Upon empirically testing our optimally weighted RandomForests learning model, we were able to arrive at the following values for the aforementioned evaluation metrics:

TABLE IV: Result Evaluation Metrics.

Accuracy	94.29%
Precision	94.5%
Recall	94.3%
F-Measure	94.3%

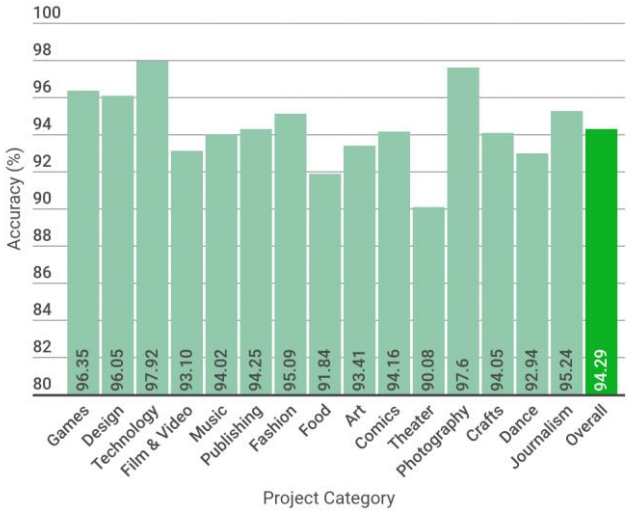

Fig. 4. Category-Wise Accuracies Using Random Forests

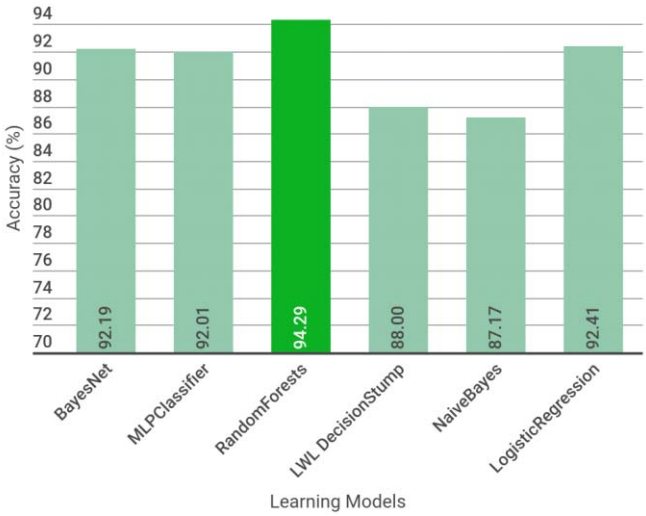

Fig. 5. Comparison of Accuracies of Different Learning Models.

We then partitioned the dataset on the basis of project category and performed the classification process on these 15 datasets, with 12 parameters. The findings of this process are illustrated in Figure 4, which shows the performance of our learning model over each of the categories. It is easy to observe that the accuracy obtained on the entire dataset is almost equal to the accuracies obtained on each dataset with a different category.

The optimally weighted RandomForests learning model eclipsed the performance of other, more conventional algorithms by at least 1 percent in all cases. Figure 5 shows the accuracies of success prediction by some well-known algorithms on the dataset, in comparison to the accuracy produced by our RandomForests learning model. It is interesting to note that even the learning models that are either commonly used or have been implemented in past literature, gave greatly improved results over the data.

We were also able to compare the importance of various predictors with respect to each other, based upon the average impurity decrease (Gini-index) and the number of nodes using that predictor. It was found that *#Websites*, which is the number of webpage links given about the project, and *SmogMain*, which is the Smog Index (readability measure) of the project description, proved to be amongst the most important predictors from amongst the 13 predictors. This was experimentally verified by attempting to perform success prediction on the dataset without the *#Websites* and *SmogMain* predictors. This time, the learning model yielded an accuracy measure of only 78.62%.

It also came to our notice that the *#created* predictor, which is the number of projects created in the past by the current project creator, had the least importance. In fact, it was not used even once to classify success in this dataset. This result conflicted with our intuition that an experienced project creator would have a better grasp of the platform and hence, find it easier

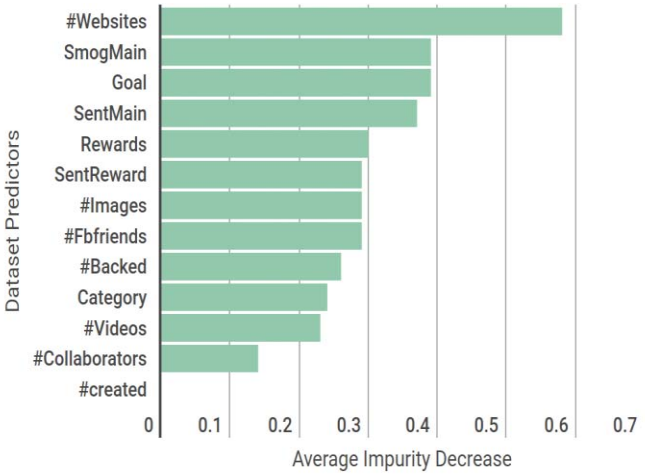

Fig. 6. Comparison of Predictor Importances.

to receive the appropriate funding for their project. Figure 6 shows the different predictors in order of their importance, as decided by their average impurity decrease in the learning model.

The limitations of our model are brought into light when trying to predict project success at various instances of its duration using temporal features. Chung and Lee [2] incorporated temporal features into their study. Our proposed model only predicts success at the inception of the project.

V. RELATED WORKS

There have been previous attempts to use a predictive model to forecast the success of a project from an online crowdfunding website such as Kickstarter, considering the popularity of crowndfunding ventures. Greenberg et al [1] in 2013 used a variety of machine learning techniques and ran classification algorithms such as SVMs, decision trees, REPTree, logistic regression etc. in order to try and classify projects as either bound to be successful or otherwise. They reached Decision Trees as their optimal learning model, with 68% accuracy in prediction of success.

Etter [3] attempted to put the time series of pledged money per project to use, to classify live projects as probable success and failures. This research also tied social features with the success prediction model and showed significant improvement in the predictive power of the model. Their learning model of choice was the Support Vector Machine (SVM).

Chung and Lee [2] improved the predictors that were existent in the model present at that time and also attempted to add temporal features and Twitter analytics, which allowed them to achieve 76.4% accuracy with static features and 78.9% accuracy with Twitter features, using the AdaBoost M1 classifier. They also created a pledged money range indicator which could predict the total funds collected per project with 86.5% accuracy.

Compared with the previous research work, we obtained the highest accuracy for the success prediction process using our modified RandomForests algorithms using optimally weighted Decision Trees, on static data. Our next step with this line of thinking would be to complement this research with real-time data, which would allow for the success prediction of live projects as well, at the time of their inception or at any point during their fundraising period.

Another comparison lies in the selection of predictors for the learning model. We are the first to use the parameters of number of websites related to the project on the page, and the readability index of the project description, together in the same dataset. Empirical testing has already shown that the absence of these predictors brings the results of the evaluation metrics down to the results obtained in past literature.

TABLE V: Comparison With Previous Works.

Decision Trees, 2013	68% (static)
SVM, 2013	76% after 4 hours from launch (real-time)
AdaBoostM1, 2015	76.4% (static), 78.9% (Twitter-augmented)
RandomForests, 2017	94.29% (static)

VI. CONCLUSION

The problem of being able to predict the success of a project on an online crowdfunding website such as Kickstarter was effectively tackled in this research, where a dataset of over 26 thousand projects trained and tested experimentally using our novel proposed approach in comparison to pre-existing learning models and algorithms. The Optimally Weighed RandomForests learning model proved to be the most effective at

success prediction, with a 94.29% accuracy rate. This research suggests that the same learning model can be used over the entire dataset, or over each category of projects. We also proposed the use of predictors that have not been used together before, such as *#websites* and *SmogMain*. Finally, we showed how the number of projects created by the creator in the past has little or no effect on the success of the current project,contrary to intuition, which serves as good news for novice creators with no experience in crowdfunding. We believe that this research serves two purposes in one shot, since it allows the project creator to make the necessary changes in their presentation of their project online, as well as allowing the project backer to determine whether a given project is worth the investment he can offer. Our future work shall be to include real-time factors to allow the success prediction of live projects as well.

REFERENCES

[1] M. D. Greenberg et al., "Crowdfunding Support Tools: Predicting Success and Failure" CHI'13, Paris, France: April 27 - May 2, 2013.

[2] J.Chung and K. Lee, "A Long-Term Study of a Crowdfunding Platform: Predicting Project Success and Fundraising Amount", HT '15, Guze- lyurt, Northern Cyprus: September 1-4, 2015.

[3] V. Etter et al.,"Launch Hard or Go Home!-Predicting the Success of Kickstarter Campaigns" COSN'13, Boston, Massachusetts, USA: October 7-8, 2013.

[4] M. Zhou et al., "Project description and crowdfunding success: an exploratory study" Information Systems Frontiers, Springer, December 7, 2016, pp. 1-16.

[5] T.K. Ho, "Random Decision Forest", Proceedings of the 3rd International Conference of Document Analysis and Recognition, Montreal, QC: 14- 16 August 1995.

[6] G. H. McLaughlin, "SMOG Grading a New Readability Formula", Journal of Reading, May 1969.

[7] S.J. Winham et al., "A Weighted Random Forests Approach to Improve Predictive Performance", Statistical Analysis and Data Mining: The ASA Data Science Journal, 2013, Volume 6, Issue 6, pp. 496-505

[8] H.B. Lee et al., "Trees Weighting Random Forests Method for Classify- ing High-Dimensional Noisy Data", IEEE 7th International Conference on e-Business Engineering (ICEBE), 2010.

[9] M. D. Greenberg, E. M. Gerber, "Learning to Fail: Experiencing Public Failure Online Through Crowdfunding", CHI 2014, Toronto, Ontario, Canada: April 26-May 1, 2014.

[10] T. Mitra, E. Gilbert "The Language that Gets People to Give: Phrases that Predict Success on Kickstarter", CSCW14, Baltimore, Maryland, USA: February 15-19, 2014.

[11] Q. Dai et al., "Research of Decision Tree Classification Algorithm in Data Mining", International Journal of Database Theory and Application Vol.9, No.5, 2016, pp.1-8

[12] L. Breiman, "Random Forests", Statistics Department, University of California Berkeley, CA 94720: January 2001

[13] Q. Ren et al., "Research on Machine Learning Framework Based on Random Forest Algorithm", Advances in Materials, Machinery, Electronics I AIP Conf. Proc. 1820, DOI: 10.1063/1.4977376

[14] A. Chandra and X. Yao "Ensemble Learning Using Multi-Objective Evolutionary Algorithms", Journal of Mathematical and Algorithms, 2006, Volume 5, Issue 4, pp. 417-445.

[15] S.G. Soares et al., "A Genetic Algorithm for Designing Neural Network Ensembles", GECCO12, Philadelphia, Pennsylvania: July 7-11, 2012.