https://doi.org/10.26599/IJCS.2022.9100003 International Journal of Crowd Science

Predicting the Entrepreneurial Success of Crowdfunding Campaigns Using Model-Based Machine Learning Methods

Michael Safo Oduro[1], Han Yu[1] ✉, and Hong Huang[2]

ABSTRACT

A common phenomenon that increasingly stimulates the interest of investors, companies, and entrepreneurs involved in crowd funding activities particularly on the Kickstarter website is identifying metrics that make such campaigns markedly successful. This study seeks to gauge the importance of key predictive variables or features based on statistical analysis, identify model-based machine learning methods based on performance assessment that predict success of a campaigns, and compare the selected different machine learning algorithms. To achieve our research objectives and maximize insight into the dataset used, feature engineering was performed. Then, machine learning models, inclusive of Logistic Regression (LR), Support Vector Machines (SVMs) in the form of Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), and random forest analysis (bagging and boosting), were performed and compared via cross validation approaches in terms of their resulting test error rates, F1 score, Accuracy, Precision, and Recall rates. Of the machine learning models employed for predictive analysis, the test error rates and the other classification metric scores obtained across the three cross-validation approaches identified bagging and gradient boosting (the SVMs) as more robust methods for predicting success of Kickstarter projects. The major research objectives in this paper have been achieved by accessing the performance of key statistical learning methods that guides the choice of learning methods or models and giving us a measure of the quality of the ultimately chosen model. However, Bayesian semi-parametric approaches are of future research consideration. These methods facilitate the usage of an infinite number of parameters to capture information regarding the underlying distributions of even more complex data.

KEYWORDS

crowdfunding; machine learning; entrepreneurship; cross validation; Support Vector Machines (SVM)

Crowdfunding is an alternative method of raising money for a project or an idea through online donations. With crowdfunding, an entrepreneur raises external financing from a large audience (the "crowd"), in which each individual provides a very small amount, instead of soliciting a small group of sophisticated investors[1]. As the internet grew tremendously in popularity in the mid-2000's so did the appearance of online crowdfunding websites. Some of these sites include Kiva, IndieGoGo, GoFundMe, and Kickstarter. Kickstarter started in 2009, based out of Brooklyn, New York and has since become one of the most popular crowdfunding websites. Since then there have been billions of dollars poured into projects from numerous backers all over the world. Kickstarter carved out its niche within the crowdfunding community as a place where creative ideas can potentially receive funding. These may fall into one of many categories: art, comics, crafts, dance, design, fashion, film/video, food, games, journalism, music, photography, publishing, technology, and theater. The amount of money that has been generated through Kickstarter has shown the world a new avenue for raising money which has piqued the interest of investors, companies, and entrepreneurs. One common question these groups have asked is: Can we identify which key factors make a Kickstarter campaign successful? Some notable Kickstarter success stories, include the Pebble Time smartwatch, which raised over 20 million US dollars (USD), the Coolest Cooler, which raised over 13 million USD, and the Exploding Kittens board game, which raised over 8 million USD. In order for a Kickstarter campaign to be deemed successful, the project needs to get hundred percent of the funding that the project founder is asking for within a set time frame (between 1 and 92 days). Otherwise it is deemed a failure. This is considered an "all or nothing" approach because the backers receive their money back if a campaign is unsuccessful. This business model encourages founders to set realistic goals and helps to protect the backers[1]. The concept of reward-based or donation-based crowdfunding entails contributors receiving token rewards or non-monetary compensation for their financial contributions. This compensation is in direct proportion to the contributions made[2]. Crowdfunding happens online on a variety of websites. There are hundreds of crowdfunding and fundraising websites with varying characteristics that meet clients' campaign goals. Understanding the unique features of these websites is critical to successful crowdfunding. Of the types of crowdfunding campaigns, it has been observed from past studies that the reward-based type is particularly appealing to potential funders[3,4]. Notable among these reward-based crowdfunding websites are Kickstarter, GoFundMe, and IndieGoGo. This research is based on the Kickstarter crowdfunding website, which is one of the world's most prominent reward-based crowdfunding platforms. It hosts funding campaigns for varying creative projects, such as arts, music, technology, films, and games. Kickstarter projects usually have a clearly defined goal. In general, the crowd funding model consists of three types of actors: the creators who propose projects

1 Department of Applied Statistics and Research Methods, University of Northern Colorado, Greeley, CO 80639, USA
2 School of Information, University of South Florida, Tampa, FL 33620-9951, USA
Address correspondence to Han Yu, han.yu@unco.edu

to be funded, backers who pledge money to back the initiator's idea, and a mediator. The Kickstarter platform mobilizes both parties. It is open to creators and backers from many countries in the world. In fact, since its inception in the year 2009, Kickstarter has hosted over 170 000 successfully funded projects raking taking in over 4.5 billion dollars from over 16 million backers. Kickstarter operates on an "all-or-nothing" funding system; this means that no one is charged for a pledge towards a project unless it reaches its funding goal and by so doing poses less risk for everyone involved. Every project consists of a target funding limit/goal over a fixed period of time; a project is considered to be a success only if this goal is met. If projects do not reach their funding goal, creators do not receive any of the pledged amount and are not obligated to complete projects without the funds required to do so, and backers will not be charged. Once a project is successfully funded, Kickstarter deducts a 5% fee from the funds solicited from the campaign. This marker of the success or failure of a campaign enables researchers to apply classification algorithms. Prospective participants (creators and backers) are usually interested in knowing the probability of success of Kickstarter campaigns to be able to achieve their goal. This potentially insulates them from investing time and money on projects that have little to no chance of being funded and, most importantly, direct them to projects with more successful prospects. A successful crowdfunding campaign can be attributed to a few factors[5], such as developer credibility and prior experiences[3]. Variables, such as the content of the campaign, financial incentives, developer and sponsors' characteristics, feedback perspective, duration of campaign, deadline, goal expectation, and precision of information provided were investigated for the crowdfunding success[6–8]. However, determining which variables are critical is difficult. This study presents a case study in which feature selections and compared respective statistical models are used to assess successful crowdfunding predictions, shedding light on prediction model selection and optimization in crowdfunding success. More specifically, the aim of this study is to analyze and compare working models that can successfully predict Kickstarter campaigns, gauge the importance of key variables or features, such as Backers Count and amount in USD pledged, and to compare different machine learning algorithms, including Logistic Regression (LR), Support Vector Machines (SVMs) which are inclusive of linear and quadratic discriminant analysis, and ultimately random forests (bagging and boosting).

1 Data Description and Feature Engineering

The data used in this study result from crowdfunding campaigns conducted on the Kickstarter website between 2009 and 2017. The data was scraped in its original form by web robots. Projects with missing observations were removed from the original data so the data was inclusive of only those projects which had reached their specified time so as to have a distinct marker of outcome: success or failure. The resulting data without missing observations consisted of 82 228 projects with information recorded on 21 features. Notable among the features considered were: country from which campaign was launched; goal/amount targeted; amount pledged over time; number of backers or backers count; project category (including art, design, food, games, movie, music, photography, publishing, and technology); amount pledged in USD; amount pledged per person; percent of goal achieved; length of Kickstarter; state from which campaign is launched; backers as a percentage of population; days spent making the campaign; days from inception to deadline; response denoting success or failure; time and population factors categorized as short, medium, and long; and other features.

1.1 Feature engineering

To maximize insight into the dataset, feature engineering was performed. Summary statistics obtained from the data showed that 36 959 projects were considered successful, representing 45.95% of the total, and the remaining 45 269 were considered failures, representing 55.05%. The projects originated from 19 countries, with the majority of projects launched in the United States (about 96%) (see Table 1). Further descriptive statistics revealed that the state of California had the most projects (12 906) and Delaware had the least (49). It was also observed (see Table 2) that music projects seem to have been the most successful followed closely by art and technology projects. Photography projects however were the least successful. A population factor was created by identifying cities with population size less than 93 794, between 93 794 and 1 211 704, and greater than 1 211 704. These were classified as low, medium, and highly populated cities, respectively. It is observed from the side-by-side bar chart in Fig. 1 that the projects from highly populated cities are more likely to be successful than those from less populated cities.

Kickstarter advises stakeholders that projects lasting 30 days or less tend to have higher success rates. Hence, having projects successfully funded in time is very crucial to project creators, not only raising the initial funds to get the project ideas off the ground, but also gaining exposure and helping them to get attention to other potential investors. As observed in Fig. 2, if the number of days from the launch of project to deadline is less than or equal to 30 days, the project tends to be successful. Since the number of Kickstarter campaigns launched was relatively higher for the United States than all other countries, our analysis is restricted to the projects in this country. In fact, for the US data, it was realized that 35 337 projects were marked successful in contrast to 34 466 being unsuccessful after "data cleaning" was performed. Stacked plots for the US dataset in Fig. 3 seem to tell a similar story to the full dataset.

Some interesting trends and patterns were further observed in the data. A variable of interest, percentage of goal (Prct_goal),

Table 1 Summary statistics on country by KickStarter project.

Status	AUS	AUT	BEL	CAN	CHE	DEU	DNK	ESP	FRA	GBR	IRL	ITA	MEX	NLD	NOR	NZL	SGP	SWE	USA
0	124	9	19	249	18	116	20	52	53	550	19	97	83	59	9	25	35	33	43699
1	128	5	11	290	17	72	40	33	77	704	20	25	37	36	16	31	23	27	35367

Table 2 Summary statistics on category by KickStarter project.

Status	Art	Design	Food	Games	Movie	Music	Photography	Publishing	Technology
0	5 701	1 383	2 248	1 355	4 114	8 997	1 415	12 302	7 754
1	3 880	2 415	815	2 165	4 530	11 846	707	7 720	2 881

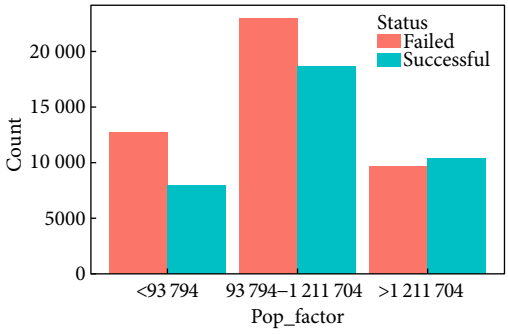

Fig. 1 Bar chart representing classification of Kickstarter projects of backer's count by population (Pop_factor) and success status.

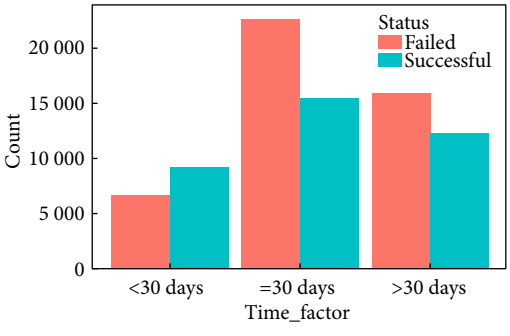

Fig. 2 Bar chart representing classification of Kickstarter projects of backer's count by time (Time_factor) and success status.

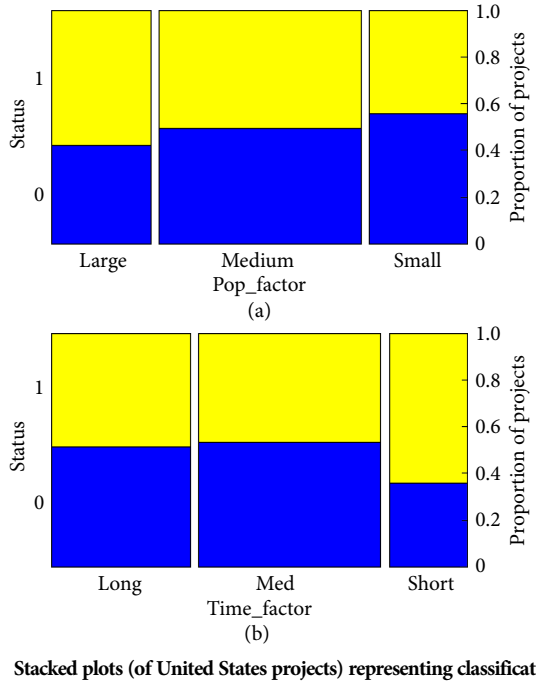

Fig. 3 Stacked plots (of United States projects) representing classification of Kickstarter projects of status by population factor (a) and status by time (b). The colors represent status that yellow means 1 (success) and blue means 0 (failure).

follows a bimodal distribution (Fig. 4) with excess zeros and excess successes (100 percent funded). Two variables that may be important predictors for determining Kickstarter success are the amount pledged and the backer count (Backers_count). The histograms that follow show the truncated distributions of both pledged USD (Fig. 5) and the backer's count (Fig. 6). The histograms are truncated at the 3rd quantile due to the extremely long right-tail. Both histograms display a similar distribution with

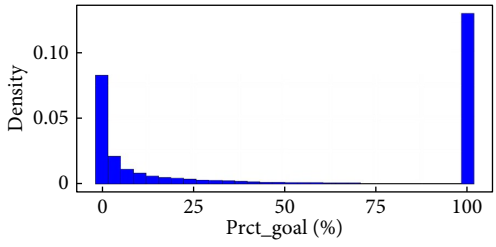

Fig. 4 Histogram display of the distribution of goal percentage.

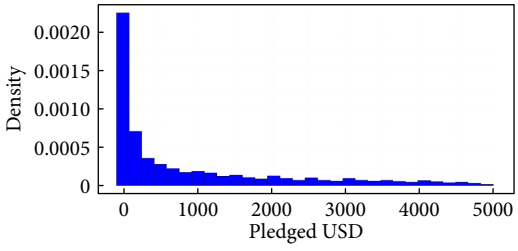

Fig. 5 Histogram display of the distribution of pledged USD.

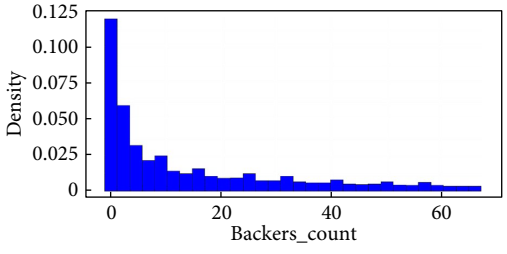

Fig. 6 Histogram display of the distribution of backer's count.

most of the values lying at or near zero with long skew right-tails. There are 15 project categories including art, comics, crafts, dance, design, fashion, film/video, food, games, journalism, music, photography, publishing, technology, and theater that are considered in the study. Certain categories tend to have higher rates of success, such as design, comics, and dance, while categories like journalism, food, and crafts tend to fail more often. All categories' average percents of goal are shown in Fig. 7, The number of days the Kickstarter accepted donations has an irregular distribution, with most campaigns lasting 30 days. This is most likely due to the recommendation that a Kickstarter campaign be 30 days or less.

The amount of money that a Kickstarter project needs to earn to be deemed successful is reflected by goal. These values are chosen by the founders when they are setting up their campaigns. These amounts range from 1 dollar up to 100 million dollars. The median value is 5 000 dollars. Figure 8 shows how the goal is distributed, although it is truncated at the 95th percentile (60 000 dollars) due to the extremely long right-tail. A variable that takes on values 0 through 4, called "Twords", was created. These are based on the most common words used in the titles and blurbs of successful campaigns. A value of 0 means that none of the words appeared in the Kickstarter's title or blurb while a 4 indicates the most appearances of successful keywords. This variable was created by looking at the name and blurbs associated with the top 10 percent of successful campaigns. The 50 most common words not including "the", "and", "it", etc., were viewed for both name and blurb. The name is the name of the Kickstarter campaign and the blurb is a short description that details further information about the Kickstarter. If any of the top 50 words was present in the title then it would be given a value of 1 or 2, depending on

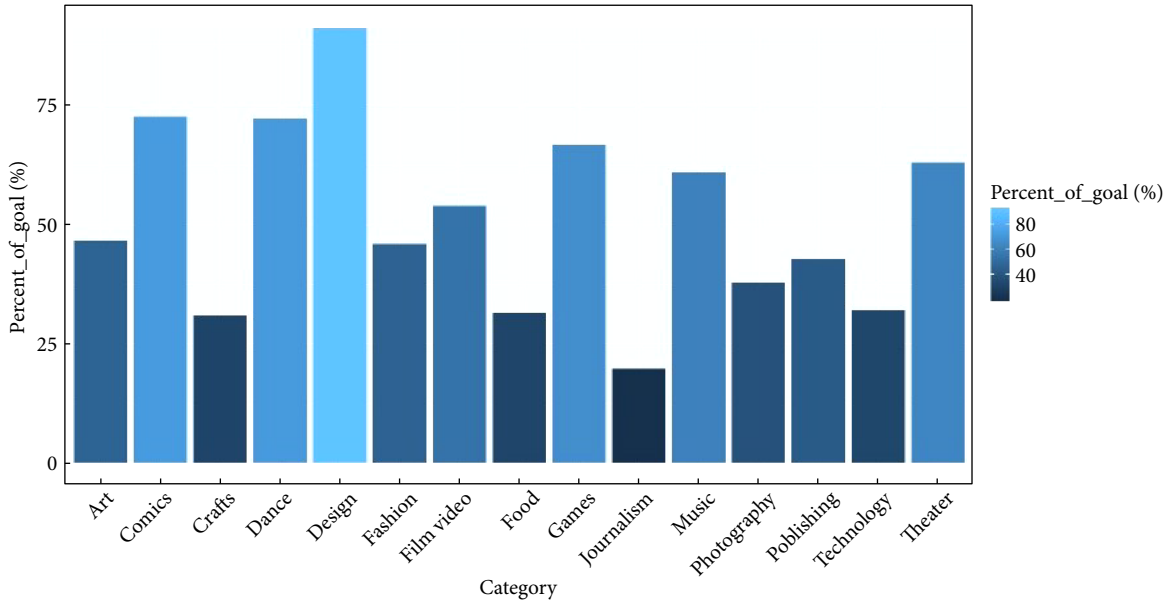

Fig. 7 Histogram display of the distribution of average percent of goal for each category.

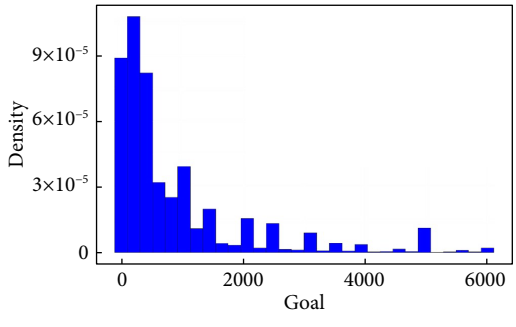

Fig. 8 Histogram display of the truncated distribution of goal.

whether the word appeared in the title once, or more than once (1 corresponding with one appearance and 2 corresponding with more than one appearance). The same values were assigned for blurb, following the same rules as the assignments for title. "Twords" was created by summing these two values together, thus, it takes on the values from 0 through 4. Figure 9 represents a stacked plot display of the frequency failed and successful projects. The plot seems to indicate that design, dance, comics, theater, and game projects are markedly successful on the Kickstarter platform as evidenced by their higher success rates. In contrast, the plot suggests a very poor performance for journalism and craft related projects.

The population of the city where the Kickstarter was launched was a variable explored. These values range between 1 231 and 8 107 916. The median city population is 422 908, with a mean of 1 233 572. States were also examined, with many states show differing levels of success, as shown in Fig. 10.

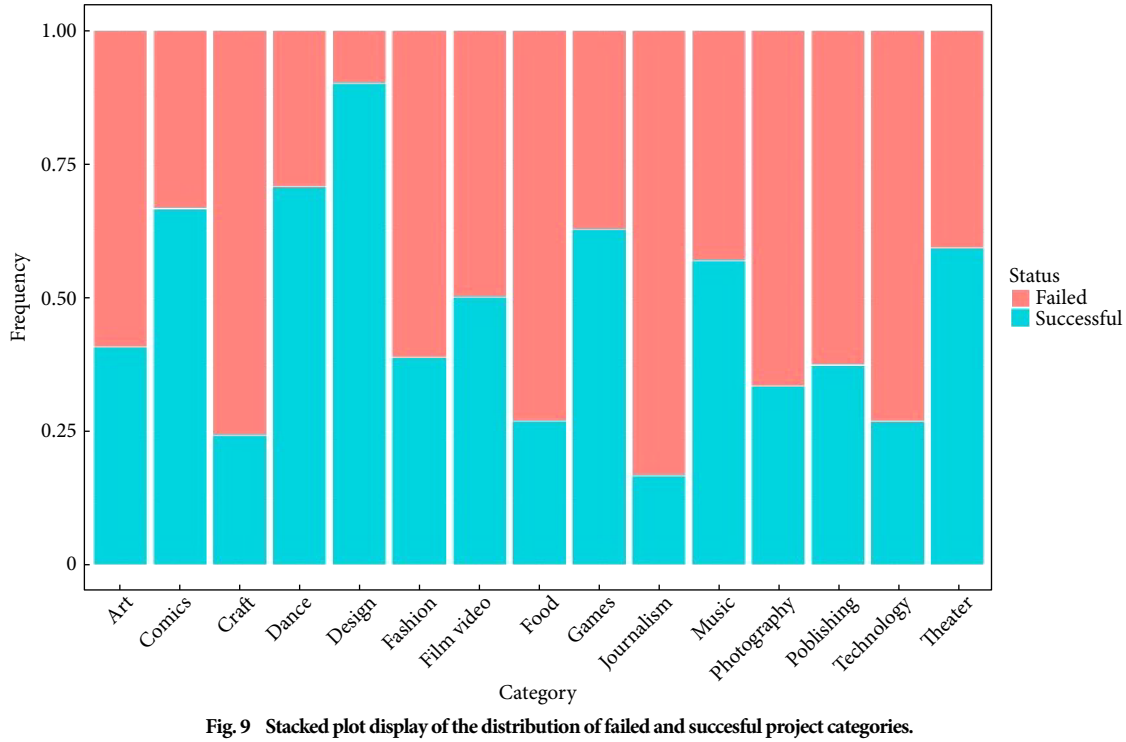

Fig. 9 Stacked plot display of the distribution of failed and succesful project categories.

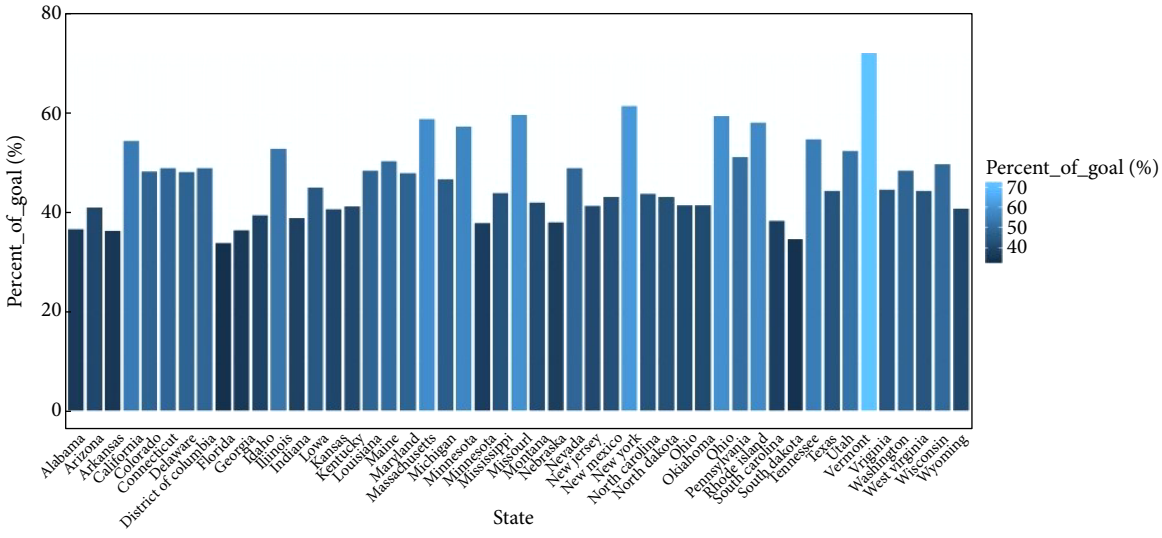

Fig. 10 Histogram display of the average percent of goal for different states.

1.2 Feature and variable selection

An attempt is made to establish possible relationships between continuous variables in the dataset. To achieve this, a correlation plot (see Fig. 11) was obtained for several selected variables. A closer look at the plot reveal highly positive correlations between some continuous variables. For example, "pledgedUSD" and "pledged" are highly correlated. This makes sense as these variables contain very similar information. The same could be said of "days_spent_making_campaign" and "days_inception_to_ deadline", and several other continuous variables. It is important to note that the presence of high correlation between these variables is an indicator of multicollinearity and may result in unreliable statistical inferences. To identify multicollinearity issues and address them, a so-called Variance Inflation Factor (VIF), condition indices, and variance decomposition proportions are used as detection measures. The VIF for each term in the model measures the combined effect of the dependences among the regressors on the variance of that term[9]. One or more large VIFs indicate multicollinearity.

Practical experience indicates that if any of the VIFs exceeds 5 or 10, it is an indication of multicollinearity. Furthermore, condition indices greater than 30 and variance decomposition proportions greater than 0.5 are recommended guidelines for detecting multicollinearity. First, the VIF, condition indices, and variance decomposition proportions of the variables are obtained

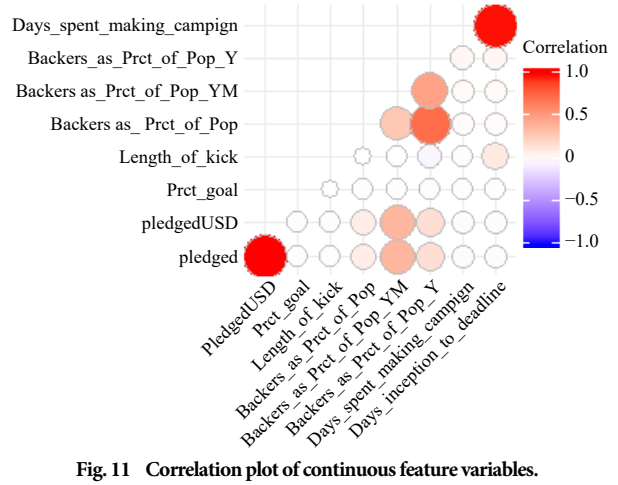

Fig. 11 Correlation plot of continuous feature variables.

"cursorily" by means of a linear model. Results regarding the VIF and variance decomposition proportion measures on the continuous variables "goal", "backers count", "pledge per person", and " length of Kickstarter" facilitated the removal of the other continuous variables. In the presence of very large amounts of data with numerous potential technical predictors, such as that used in this Kickstarter project, it is infeasible for investigators or researchers to put all the potential predictors into a model, as many of these variables may not be associated with the outcome being predicted. In these scenarios, one may be interested in the prediction of an outcome and finding a "parsimonious" subset of variables that are associated with the outcome. This means that we can find a dimension reduction technique or method to determine the most important variables for analysis. In our case, we consider the use of the Least Absolute Shrinkage and Selection Operator (LASSO), which can assist investigators interested in predicting an outcome by selecting the subset of the variables that minimizes prediction error[10]. Here, the coefficients of some less contributive variables are forced to be exactly zero. Only the most significant or contributive variables are kept. The random forest approach or the criterion called Gini Importance or Mean Decrease in Impurity (MDI) that calculates the importance of each feature also presents us with a variable importance measure[11]. When both methods were applied, the variables goal, backers count or number of backers, pledge per person, length of Kickstarter project, project categories, time factor, and population factor were ranked as more contributive variables or the most significant variables in minimizing prediction error.

2 Methods

2.1 Classification algorithms

In this section, the machine learning algorithms explored in identifying the best predictive model for the Kickstarter data are explained. The classification algorithms employed are LR, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), classification trees, bagging, and boosting. Validation methods and the results of these methods are also reported.

2.1.1 LR

The LR model is a binary classification model for supervised learning in machine learning. In the LR model, the binary response follows a binomial distribution with probability of

success π and probability of failure $1-\pi$ under the assumption that there are n independent and identically distributed Bernoulli trials; that the number of trials are fixed and that there are two and only two outcomes, labelled success and failure. This classification model models the probability of success as the conditional expected value of the response variable given the features x, that is $\pi(x) = E(Y \mid x)$ with the logit link function to the predictor,

$$\alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_p x_p,$$

where $\alpha_1, \alpha_2, \ldots, \alpha_p$ are coefficient parameters of the features.

The LR model is given by

$$\text{logit}\,[\pi(x)] = \log\left[\frac{\pi(x)}{1-\pi(x)}\right] = \alpha_0 + \alpha_1 x_1 + \alpha_2 x_2 + \cdots + \alpha_p x_p.$$

That is

$$\pi(x) = \frac{e^{\alpha_0+\alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_p x_p}}{1+e^{\alpha_0+\alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_p x_p}},$$

which can take the range of values from 0 to 1. The likelihood function of the LR model is

$$L(\alpha \mid y) = \prod_{i=1}^{n} [\pi(x_i)]^{y_i} [1-\pi(x_i)]^{(1-y_i)},$$

$$L(\alpha \mid y) = \prod_{i=1}^{n} \left[\frac{e^{\alpha_0+\alpha_1 x_1+\cdots+\alpha_p x_p}}{1+e^{\alpha_0+\alpha_1 x_1+\cdots+\alpha_p x_p}}\right]^{y_i} \times \left[1-\frac{e^{\alpha_0+\alpha_1 x_1+\alpha_2 x_2+\cdots+\alpha_p x_p}}{1+e^{\alpha_0+\alpha_1 x_1+\cdots+\alpha_p x_p}}\right]^{1-y_i},$$

where $y_i = 0$ or 1. For maximum likelihood estimation, this function can be maximized by taking the natural logarithm of the likelihood function, differentiating with respect to the parameters, equating to zero, solving the equations using the iterative least squares method and obtaining $\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_p$ [12].

2.1.2 LDA

Although the LR model is a relatively powerful yet simple linear classification algorithm, it has limitations that necessitates the need for alternate linear classification algorithms. For example, when the two response classes are well-separated, the parameter estimates of this model become very unstable. Furthermore, for relatively small sample sizes, when the distribution of the features in the model is Gaussian distributed, the LDA becomes more stable than the LR model. LDA essentially models the distribution of features separately in each response class and then adopts Bayes theorem to estimate probabilities. LDA makes predictions by estimating the probability that a new set of features belong to each class. The class that gets the highest probability is the output class and a prediction is made. More intuitively, LDA can be derived from probabilistic models that model the conditional distribution of the data for each class $c, P(X \mid Y = c)$. LDA assumes that each data class follows or is modeled by a multivariate Gaussian distribution,

$$f_c(X) = P(X \mid Y = c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}} \times \exp\left(-\frac{1}{2}(X-\mu_c)'\sum_c^{-1}(X-\mu_c)\right),$$

where d represents the number of features in the model. The covariance matrix $\sum_c$ is the same across all the classes, that is $\sum = \sum_c$. LDA is assumed as a classifier, and its use is evidenced by the usage of the class priors estimated from the training data. This is done by finding the prior probabilities,

$$P(X \mid Y = c),$$

which is computed as proportions of data in each class c. The class means, μ_c as well as the covariance matrix $\sum$ are estimated by prior probabilities,

$$\hat{\pi}_c = \frac{n_c}{n}.$$

Class means is

$$\hat{\mu}_c = \frac{1}{n_c}\sum_{i:y_i=c} X_i.$$

Covariance matrix is

$$\Sigma = \frac{1}{n-C}\sum_{c=1}^{C}\sum(X_i - \hat{\mu}_c)^2.$$

In general, the classification function prescribed for new data points is as below:

$$q(x) = \underset{c\in\mathbf{R}}{\text{argmax}}\ P_{x|y=c}(x \mid Y = c)P(Y = c).$$

In the case of a binary classification as with our Kickstarter problem, $Y = \{1, 0\}$, the classification function is then represented as

$$Y = \begin{cases} 1, & \text{if } P(X \mid Y=1)P(Y=1) \geqslant P(X \mid Y=0)P(Y=0); \\ 0. \end{cases}$$

The general LDA classification function is

$$F(x) = \text{argmax}_c\, \delta_c(x),$$

where

$$\delta_c(x) = x'\sum_c^{-1}\mu_c - \frac{1}{2}\mu_c'\sum_c^{-1}\mu_c + \log \pi_c.$$

2.1.3 QDA

The LDA models the binary response with a linear combination of the features. The QDA is similar to LDA in terms of the derivation of parameters. However, the underlying difference is that QDA models/classifies the response with a non-linear combination of features. Furthermore, unlike the LDA classifier, QDA assumes that each class of the training data possesses its own covariance matrix. This means that an observation pertaining to the c-th class will be of the form $X \sim N(\mu_c, \Sigma_c)$, with its own class covariance matrix Σ_c. The decision boundary between the two classes is quadratic rather than a hyperplane. The QDA discriminant function is

$$\delta_c(x) = -\frac{1}{2}\log\left|\sum_c\right| - \frac{1}{2}(X-\mu_c)'\sum_c^{-1}(X-\mu_c) + \log \pi_c.$$

QDA estimates a covariance matrix for each class, and hence the number of effective parameters are greater than LDA. In terms of flexibility, LDA is a relatively better classifier, but if the training observations are very large as in our case, then the use of a QDA for classification is plausible.

2.2 Tree-based methods

Tree-based methods in machine learning are popular algorithms for classification and regression. These methods are notable in terms of their high prediction accuracy, stability, and their ease of interpretation. Furthermore, they are robust for investigating non-linear relationships. Tree-based methods involve segmenting the feature space into regions. In terms of prediction, the summaries

of the training observations are used: that is, the mean and the node. There are so-called splitting rules used to segment the feature space. One merit of tree-based methods is their non-parametric nature; they have no underlying distributional assumptions about their feature space and the classifier structure. The tree-based methods employed in this project are classification trees, Bagging, and Boosting.

2.2.1 Classification trees

Classification trees are a type of decision tree algorithm. They are used for the prediction of the membership of observations into classes of a categorical response from measurements taken on features. The idea behind the prediction is that each observation belongs to the most commonly occurring class of the training observations in the region to which it belongs. A classification tree is comprised of branches that represent attributes and leaves that represent decisions. In practice, the decision process commences at the trunk and follows the branches until a leaf is reached. For a classification tree algorithm, the interest is in class prediction of class proportions among training observations in their respective regions as well as class predictions corresponding to specific terminal node regions. The algorithm is an embodiment of the concept of recursive binary partitioning or splitting. This involves dividing up the dimensional space of the features into nonoverlapping rectangles. This division is accomplished recursively. The criterion used in making those binary splits is the so-called classification error rate, which is the proportion of incorrectly classified training observations in a region that do not belong to the most common class. To define this classification error rate, also known as the misclassification error rate, we need to define the proportion. For a node s, which represents a region B_s with N_s corresponding observations, the proportion of class c observations in node s observations is represented as

$$\hat{p}_{sc} = \frac{1}{N_s} \sum_{x_i \in B_s} I(y_i = c).$$

The majority class for node s is represented as $c(s) = \text{argmax}_c \hat{p}_{sc}$ and hence the misclassification error can be written out as

$$E = \frac{1}{N_s} \sum_{x_i \in B_s} I(y_i \neq c(s)) = 1 - \hat{p}_{sc}.$$

Alternatively, two other measures that are used in place of the misclassification rate are the so-called Gini Index and the cross-entropy rate. The Gini Index is the measure of the total variance across the classes and sometimes described as the measure of node purity.

The Gini Index is represented as

$$\sum_{c \neq c'} \hat{p}_{sc} \hat{p}_{sc'} = \sum_{c=1}^{c} \hat{p}_{sc} (1 - \hat{p}_{sc}).$$

The cross entropy is defined as

$$-\sum_{c=1}^{c} \hat{p}_{sc} \log \hat{p}_{sc}.$$

2.2.2 Bagging

Bootstrapping is an increasingly popular and powerful concept that is used in machine learning. It simply refers to a resampling algorithm used to estimate statistics such as standard errors, means, and variances from a population by randomly resampling a dataset with replacement. The bootstrap facilitates understanding of the biases, variances, and features that exist in the resample and its application spreads to a variety of statistical learning methods, including those whose measure of variability is difficult to estimate. In essence, this method can be useful for testing the stability of a model, as multiple datasets are resampled, used, and tested on multiple models.

The aggregated bootstrap, or bagging, is an ensemble method which is an extension of the bootstrap method in matching learning that is applied to decision trees that suffer from very high variance. Decision trees generally suffer from high variance as splitting training observations/datasets randomly and fitting classification/regression trees to these random datasets may yield completely different inferences. Bagging comes to the rescue, as it can reduce the uncertainty associated with fitting decision trees with the randomly split datasets. Essentially, bagging reduces the variance associated with decision trees. From a training dataset, what bagging does is by using the bootstrap method, it repeatedly samples without replacement and generates G different bootstrapped training datasets. Different prediction models are fitted using the independent bootstrapped datasets. Each prediction model suffers from a very high variance but low bias, especially for decision trees but subsequently all prediction models are averaged together to obtain a low variance prediction model. This "bagged" model is represented as

$$\hat{f}_{bag}(x) = \frac{1}{G} \sum_{g=1}^{G} \hat{f}^{g}(x).$$

2.2.3 Boosting (gradient boosting)

Boosting is another machine learning algorithm that reduces the variance resulting from the decision tree algorithm. It works in a similar way as bagging, except that with boosting, decision trees are grown in a sequential manner: that is, each decision/classification tree is grown from using information from previously grown classification trees. Each new tree results from the fit of a modified version of the original dataset. Unlike the bagging algorithm, boosting does not involve bootstrapping. The gradient boosting algorithm is a type of boosting algorithm for classification trees that we employ in this project. It trains predictive models in a gradual, additive and sequential manner. It discriminates the shortcomings of decision trees by using gradients in the loss function of the predictive models. The kind of desired loss function, $L(y,f(x))$, needs to be specified before hand. A modified general algorithm for the gradient tree boosting algorithm[13] is as follows.

(1) Initialize the optimal constant model, which is a single terminal node tree,

$$f_0(x) = \text{argmin}_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma).$$

(2) For $g = 1$ to G (iterations):

(a) For $i = 1, 2, \ldots, N$, compute

$$r_{ig} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f=f_{g-1}}.$$

These are referred to as pseudo/generalized residuals.

(b) Fit a regression tree to the targets r_{ig} giving terminal regions, $R_{ig}, j = 1, 2, \ldots, J_g$.

(c) For $j = 1, 2, \ldots, J_g$, compute

$$\gamma_{jg} = \underset{\gamma}{\operatorname{argmin}} \underbrace{\sum_{x_i \in R_{jg}}^{N} L\left(y_i, f_{g-1}\left(x_i\right) + \gamma\right)}.$$

(d) Update

$$f_g(x) = f_{g-1}(x) + \sum_{j=1}^{J_m} \gamma_{jg} I\left(x \in R_{jg}\right).$$

(3) Output $\hat{f}(x) = f_G(x)$.

For gradient boosting classification algorithms, a loss function that can be assumed is a multinomial deviance. In this case, K least squares trees will be constructed at each iteration. Each tree, T_{kg} will be fitted to its negative gradient h_{kg},

$$-h_{ikg} = \frac{\partial L\left(y_i, f_{1g}\left(x_i\right), \ldots, f_{lg}\left(x_i\right)\right)}{\partial f_{kg}\left(x_i\right)}.$$

Furthermore, a boosting classification algorithm will have lines 2(a) –2(d) in the algorithm repeated K times at each iteration g and will have a variant of the final output result in (3), as $\hat{f}(x) = f_{kG}(x), k = 1, 2, \ldots, K$.

2.3 Evaluating machine learning models

After exploring the machine learning models presented in Section 2, it is important to find metrics that quantify the performance of the predictive models. There are several metrics that are available for evaluating varying machine learning tasks. In this article, we focus on cross-validation approaches and classification metrics for evaluating our models. These metrics are inclusive of test error rates, Accuracy, Precision, Recall, and an F-measure (also sometimes known as the F1 score).

2.3.1 Cross-validation: Validate set approach and *k*-fold cross-validation

Cross-validation involves estimating the test errors associated with the algorithms considered to be able to evaluate their performance. A good cross-validation method will give a robust measure of the various predictive models' performance throughout the whole dataset. The two cross-validation approaches considered in this article are the validation set approach and the k-fold validation approach. The validation set approach, also known as the hold-out validation set approach, involves splitting the available set of observations into two non-overlapping parts, called a training set and a test set (or hold-out set). For this project, the data split was 70% of the data for training and 30% of the data for testing. The predictive models of the various algorithms are fitted to the training set and the fitted models are used to predict observations for the test set. We can then obtain classification test error rates for model evaluation. The merit of the validation set approach is its simplicity in terms of implementation and its low computational complexity. However, the downside of this method is that it may suffer from issues of high variance. This is a result of the uncertainty resulting from which observations will end up in either the holdout set or training set. Hence the result may be different for different sets. The k-fold cross-validation is the next measure employed for model assessment. It involves the observations being first randomly split into k groups or folds. The first group will be used as the test set, and the algorithm is fitted to the $k-1$ remaining groups. The test error rate is then computed for the observations in the test set. There is then an iteration of the procedure k times. For each of the k times, a different group will be treated as the test set. As a result, there will be k test error estimates of the test sets and thus a reasonable approach will be to average the classification test errors to get one estimate of the test error. In this article, the 5- and 10-fold validation approaches are considered. The merit with this method is its accurate estimation performance. The higher the value of k chosen, the less biased model the method results.

2.3.2 Classification metrics

Classification metrics for evaluating predictive models are usually premised on a confusion matrix[14]. This matrix, when constructed, specifies the number of test cases that are correctly and incorrectly classified, and entails the needed information for constructing the metric. In this article, we adopt the Accuracy, Precision, Recall rate, and an F1 score as comparison metrics to the test error rates realized from the cross-validation approaches. The Accuracy metric can be defined as the number of the number of test cases correctly classified, that is, the sum of True Positive (TP) and True Negative (TN) cases divided by the total number of all test cases which also includes False Negatives (FN) and False Positives (FP). This is represented as

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}.$$

More precisely, the Accuracy metric involves an overall measure of how correctly the classification model predicts the entire dataset. Owing to its relatively easy computation and understanding, the Accuracy measure is widely used. However, a drawback with this measure is that, for highly unbalanced datasets, it masks classification errors for classes with few cases and thus, may perform poorly[15]. Another metric that is useful is the Precision. The Precision is a ratio of true positive cases predicted to the sum of TP and FP. Intuitively, this metric measures the ability to correctly detect or classify cases belonging to the positive class. The higher the ratio, the better the precision of the classification model. This is represented as

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}},$$

The Recall rate is another classification metric that is employed in this article. It is defined by the ratio of TP cases to the sum of TP and FN cases (that is, the total number of positively classified cases). Thus, this metric is informative in part, because it specifies the number of positive cases correctly predicted from the total number of positive cases. It is worthy of mention that the Precision and Recall metrics form the building blocks of the F1 score metric which is the last section considered. The Recall metric is given as

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}.$$

Finally, the last metric we consider for model comparison is the F1 score. This is a combination of the Precision and Recall metrics via a harmonic mean equation which is given as

$$\text{F1 score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} = 2 \times \frac{1}{(\text{Precision})^{-1} + (\text{Recall})^{-1}}.$$

The larger the F1 score, the higher the Precision and the Recall. It serves as a good compromise between Precision and Recall and tends to work well with highly imbalanced datasets, unlike the Accuracy metric.

3 Results

The results of the six machine learning algorithms used for prediction and their corresponding test error rates, Accuracy, F1 score, Precision, and Recall rates resulting from the cross-validation approaches are tabulated and shown in Table 3. Of the 6 methods used for prediction, the test error rates obtained across the three cross-validation methods suggest bagging and gradient boosting are the most robust methods for predicting the success of Kickstarter projects. The test error rates for linear and quadratic discriminant analysis seem to be close in comparison. in fact, the misclassification rates are around 30% for both methods. The LR model seems to come close as the next better predictive model after bagging and the gradient boosting algorithms as evidenced by its low test error rates of about just 5% –6%. These results, interestingly, are also in line with the Accuracy metric considered. Overall, the LR, bagging and gradient boosting models have relatively higher Accuracy rates and thus have better predictive performance. This is further evidenced by their F1 score, Precision, and Recall scores, which also are the higher amongst all models considered. The LDA, QDA, and tree models perform similarly but possess less Accuracy and F1 scores than their counterpart models.

4 Discussion and Future Work

This study sought to mainly investigate statistical learning methods and associated machine learning algorithms based on feature engineering that present us with the best predictive models for predicting the success of Kickstarter campaigns. The data used was web-scraped from KickStarter, one of the biggest reward-based crowd funding platforms in the world. Over 80 000 observations and 61 features were used. Because a lot of the Kickstarter projects (about 96%) originated in the United States, the emphasis of the study was placed on these projects. First, feature engineering was performed to target the most relevant variables. After a dimensionality reduction was performed with LASSO, the random forest procedure, and multicollinearity diagnostics, the variables of goal, backers count, time, and population were ranked as the most contributive and significant variables in minimizing the prediction error of any machine learning methods we planned to use. Six machine learning algorithms were then explored. The performances of these methods were employed for validity with three cross-validation approaches and classification metrics, such as test error rates, Accuracy, F1 scores, Precision, and Recall were tracked. The results showed the bagging and gradient boosting methods for classification as having the least test error rates and overall very

Table 3 Results of predictive models based on model evaluation metrics

Method: VSA	Accuracy	F1 score	Precision	Recall	Test error rate
LR	0.9371	0.9361	0.9347	0.9375	0.0622
LDA	0.6501	0.6429	0.6364	0.6494	0.3420
QDA	0.7111	0.6660	0.5986	0.7504	0.2826
Classification trees	0.8881	0.8852	0.8743	0.8964	0.0974
Bagging	0.9999	0.9999	0.9998	1.0000	0.0076
Boosting	0.9774	0.9754	0.9533	0.9985	0.0251
Method: 5-Fold	**Accuracy**	**F1 score**	**Precision**	**Recall**	**Test error rate**
LR	0.9396	0.9390	0.9400	0.9380	0.0546
LDA	0.6516	0.6445	0.6381	0.6509	0.3475
QDA	0.7111	0.6660	0.5986	0.7504	0.3008
Classification trees	0.8881	0.8852	0.8743	0.8964	0.1124
Bagging	1.0000	1.0000	1.0000	1.0000	0.0056
Boosting	0.9770	0.9751	0.9533	0.9979	0.0256
Method: 10-Fold	**Accuracy**	**F1 score**	**Precision**	**Recall**	**Test error rate**
LR	0.9396	0.9390	0.9400	0.9380	0.0546
LDA	0.6500	0.6428	0.6364	0.6494	0.3477
QDA	0.7111	0.6660	0.5986	0.7504	0.3010
Classification trees	0.7111	0.6660	0.5986	0.7504	0.1090
Bagging	1.0000	1.0000	1.0000	1.0000	0.0053
Boosting	1.0000	1.0000	1.0000	1.0000	0.0252

high Accuracy, Precision, and Recall rates, indicative of better classification methods for predicting success rates of Kickstarter campaigns. The major research question hence has been answered. However, it is important to note that for the very complex data, the assumptions for some classification methods, such as LR analysis that is a parametric approach, have been shown to be unrealistic and not flexible. This is because it first assumes that the sample data comes from a population that follows an identical probability distribution with a fixed number of parameters. The second assumption of independence of observations is not always plausible for complex datasets. Hence the Bayesian nonparametric approach will be a more plausible approach and worthy of future consideration to promote generalization. The Bayesian nonparametric models are more robust and valid across problems as they allow the usage of an infinite number of parameters to capture the features of the distribution underlying the complex data. Moreover, if the interest is the identification and understanding of the effect of particular variables considered on the rate of success, then causal inference models rather than curve fitting should be further explored.

Dates

Received: 28 January 2021; Revised: 12 April 2021; Accepted: 26 October 2021

References

[1] P. Belleflamme, T. Lambert, and A. Schwienbacher, Crowdfunding: Tapping the right crowd, *J. Bus. Venturing*, vol. 29, no. 5, pp. 585–609, 2014.

[2] V. Kuppuswamy and B. L. Bayus, Crowdfunding creatynamics oive ideas: The df project backers, in *The Economics of Crowdfunding*, D. Cumming and L. Hornuf, Eds. Cham, Germany: Springer, 2018, pp. 151–182.

[3] E. M. Gerber and J. Hui, Crowdfunding: Motivations and deterrents for participation, *ACM Trans. Comput.-Human Interact.*, vol. 20, no. 6, p. 34, 2013.

[4] J. S. Hui, M. D. Greenberg, and E. M. Gerber, Understanding the role of community in crowdfunding work, in *Proc. 17th ACM Conf. on Computer Supported Cooperative Work & Social Computing*, Baltimore, MD, USA, 2014, pp. 62–74.

[5] E. Mollick, The dynamics of crowdfunding: An exploratory study, *J. Bus. Ventur.*, vol. 29, no. 1, pp. 1–16, 2014.

[6] M. J. Zhou, B. Z. Lu, W. P. Fan, and G. A. Wang, Project description and crowdfunding success: An exploratory study, *Inf. Syst. Front.*, vol. 20, no. 2, pp. 259–274, 2018.

[7] N. X. Wang, Q. X. Li, H. G. Liang, T. F. Ye, and S. L. Ge, Understanding the importance of interaction between creators and backers in crowdfunding success, *Electron. Commer. Res. Appl.*, vol. 27, pp. 106–117, 2018.

[8] K. Choy and D. Schlagwein, Crowdsourcing for a better world: On the relation between it affordances and donor motivations in charitable crowdfunding, *Inf. Technol. People*, vol. 29, no. 1, pp. 221–247, 2016.

[9] H. Yu, S. H. Jiang, and K. C. Land, Multicollinearity in hierarchical linear models, *Soc. Sci. Res.*, vol. 53, pp. 118–136, 2015.

[10] S. L. Kukreja, J. Löfberg, and M. J. Brenner, A least absolute shrinkage and selection operator (LASSO) for nonlinear system identification, *IFAC Proc. Vol.*, vol. 39, no. 1, pp. 814–819, 2006.

[11] B. H. Menze, B. M. Kelm, R. Masuch, U. Himmelreich, P. Bachert, W. Petrich, and F. A. Hamprecht, A comparison of random forest and its Gini importance with standard chemometric methods for the feature selection and classification of spectral data, *BMC Bioinformatics*, vol. 10, no. 1, p. 213, 2009.

[12] P. McCullagh and J. A. Nelder, *Generalized Linear Models*. 2nd ed. London, UK: Chapman & Hall/CRC, 1989.

[13] J. Franklin, The elements of statistical learning: Data mining, inference and prediction, *Math. Intell.*, vol. 27, no. 2, pp. 83–85, 2005.

[14] M. Grandini, E. Bagli, and G. Visani, Metrics for multi-class classification: An overview, arXiv preprint arXiv:2008.05756, 2020.

[15] N. Japkowicz and M. Shah, *Evaluating learning algorithms: A classification perspective*. Cambridge, UK: Cambridge University Press, 2011.