The Computational Performance of Machine Learning Methods in The Success Prediction of Kickstarter Campaigns

Michael Safo Oduro* Han Yu†

Abstract
Crowdfunding is a large group or "crowd" of people who are interested in investment or donation activities to support project ideas financially. There are several kinds of crowdfunding such as equity, lending, rewards and donation crowdfunding. Kickstarter is one of the world's most prominent reward-based crowd funding platforms. This study exploits the use of machine learning methods to examine the latent structure of a web-scrapped Kickstarter data. The focal point of this project is to analyze this data by using machine learning algorithms to predict the success of a crowdfunding campaign using features inclusive of project category, duration in days from launch to deadline, population in city and many other features. The computational complexity and reliability of the predictive performance of these algorithms are examined on the large scale data.

Key Words: Crowdfunding, Kickstarter, Machine Learning Methods, Entrepreneurship

1. Introduction

1.1 Background and Objectives

Entrepreneurs, start-ups and several people around the world with diverse goals and interests in various disciplines are interested in financing options for their projects and frequently so at the initial stages. A plausible external financing means or funding alternative that has gained traction in recent years is the concept of crowdfunding. Simply put, crowdfunding represents a large group or "crowd" of people interested in investment or donation activities to support project ideas financially through the internet. Inclusive of crowdfunding types are equity, lending, rewards and donation crowdfunding. 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 of the contributions made (Belleflamme et al., 2015; Kuppuswamy et al., 2018). Crowdfunding happen online on various websites. There are hundreds of crowdfunding and fundraising websites with varying characteristics that meet clients' campaign goals. Understanding the unique features of these websites shall aid in taking absolute advantage of the prowess in crowdfunding. Of the types of crowdfunding campaigns, it has been observed from past studies that the reward-based type is significantly appealing to funders in crowdfunding sites (Gerber and Hui, 2013; Hui et al., 2014). Notable among these reward-based crowdfunding websites are Kickstarter, GoFundMe and IndieGoGo. This research is premised on the Kickstarter crowdfunding website, which is one of the world's most prominent reward-based crowd funding 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 crowdfunding model consists of three types of actors: the creators who propose projects to be funded, backers who pledge money to back the initiator's idea, and

*Department of Applied Statistics and Research Methods, University of Northern Colorado, Greeley, CO 80639

†Department of Applied Statistics and Research Methods, University of Northern Colorado, Greeley, CO 80639

a mediator. The kickstarter platform mobilizes both parties. The Kickstarter platform 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 in over 4.5 billion dollars from over 16 million backers. Kickstarter operates the "all-or-nothing" funding regime; this implies 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. If projects do not reach their funding goal, creators are not obliged to complete projects without the funds required to do so, and backers will not be charged. If the target funding limit is attained within the specified period of time, it is deemed successful, otherwise it is deemed unsuccessful and the creator or owner does not receive any of the pledged amount. Once a project is successfully funded, Kickstarter applies and deducts a 5% fee to the funds solicited from the campaign.

This marker of success or no success of campaign enables researchers to apply classification algorithms. Prospective participants (creators and backers) are usually interested to know 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 lower or no likelihood of being funded and most importantly direct them to projects with more successful prospects. The major objective of this project is to find metrics, variables or features that contribute to robust prediction of successful campaigns using semiparametric framework with machine learning methods. For example, does the project category influence the outcome of a campaign? What about duration in days from launch to deadline and population in the city from which campaign is launched? These and other several features are considered. Ultimately, using machine learning methods, we will seek the best prediction model that predicts success rates of Kickstarter campaigns based on certain metrics.

1.2 Data Description

The data used in this project result from crowdfunding campaigns conducted on the Kickstarter website over time from year 2009 to 2017. The data was scraped in its original form by web robots https://webrobots.io. 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 had $82,228$ 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 backer's count, project category (inclusive of 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, Backer's as a percentage of population, days spent making the campaign, days from inception to deadline, response denoting success or failure, time and population factors both categorized as short, medium and long and other features.

2. Feature Engineering

In an attempt to maximize insight into the dataset, a feature engineering was performed. Summary statistics obtained from the data showed $82,228$ Kickstarter projects considered over the period of 8 years between 2009 and 2017. $36,959$ projects were considered successful representing 45% of the total and $45,269$ considered failures representing 55.05%. The projects emanated from 19 countries with majority of projects launched in the United

States (about 96%). Further descriptives revealed the state of California having the most projects $(12,906)$ and Delaware state having the least (49). It is also observed that music projects launched seem to have been the most successful followed closely by art and technology projects. Photography projects however were the least successful.

Table 1: Summary Statistics on Country by Kickstarter Project

	Country																		
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

	Category								
Status	art	design	food	games	Movie	music	photography	Publishing	Technology
0	5701	1383	2248	1355	4114	8997	1415	12302	7754
1	3880	2415	815	2165	4530	11846	707	7720	2881

The population factor created was defined by identifying cities with population size less than $93,794$, between $93,794$ and $1,211,704$ and greater than $1,211,704$ as low, medium and highly populated cities respectively. It is observed from the side by side bar chart Figure 1 (left) below that the projects from highly populated cities are more likely to be successful than less populated cities

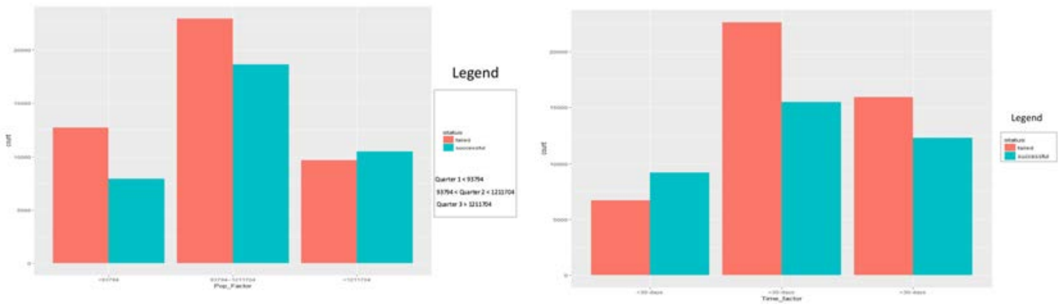

Figure 1: The bar chart for population factor (left) and time factor (right).

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 Figure 1 (right) if the number of days from the launch of project to deadline is small 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 focused on 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 Figure 2 only seems to tell a similar story as the full dataset.

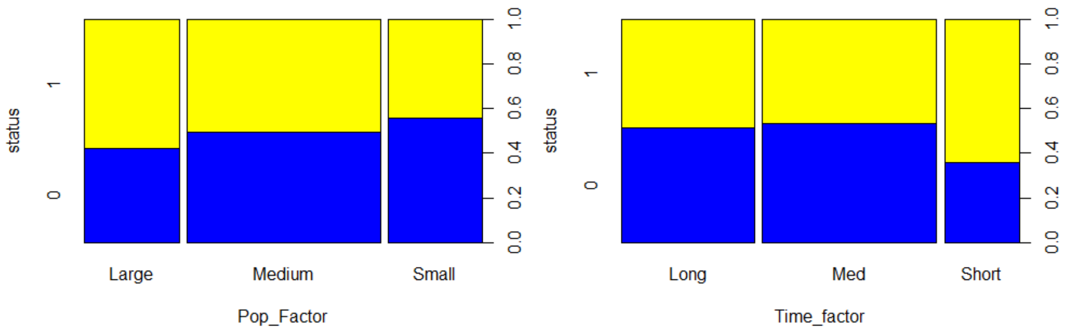

Figure 2: The bar chart for population factor and time factor.

An attempt is further made to establish possible relationships between continuous variables in the dataset. To achieve this, a correlation plot (see Figure 3) was obtained for several selected variables. A closer look at the plot revealed highly positive correlations between some continuous variables. For example, "pledgedUSD" and "pledged" are highly correlated. This makes makes sense as these variables contain very similar information. Same could be said of days spent making campaign and days from 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 (Yu et al., 2015). 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 "cursorily" by means of a linear model. Results regarding the VIF and variance decomposition proportion measures on continuous variables "goal", "backers count", "Pledge per person", and "Length of kickstarter" facilitated the removal of the other continuous variables.

2.1 Variable Selection Using LASSO and Random Forests

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 particular 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. 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 forests approach or the criterion called Gini Importance or Mean Decrease in Impurity (MDI) that calculates each feature importance also presents us with a

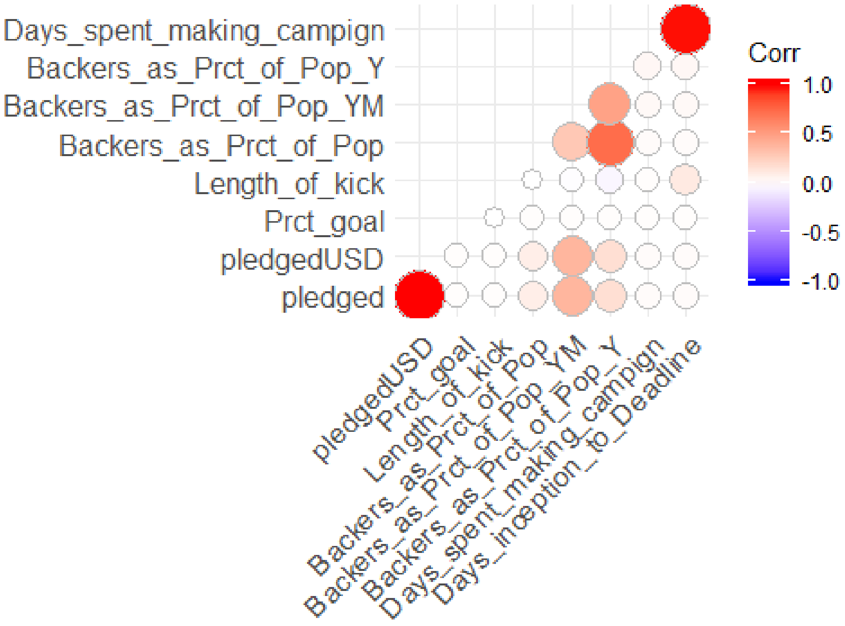

Figure 3: Correlation Plot.

variable importance measure. 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.

3. Methods and Results

3.1 Classification Algorithms

In this section, the machine learning algorithms exploited in identifying the best predictive model for our kickstarter data are explained. The classification algorithms employed are logistic regression, linear discriminant analysis, quadratic discriminant analysis, classification trees, Bagging and Boosting. Validation methods and the final results of these methods are also reported.

3.1.1 Logistic Regression

The logistic regression model is a binary classification model for supervised learning in machine learning. In the logistic regression 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 $\boldsymbol{x}$, that is $\pi(\boldsymbol{x}) = E(Y|\boldsymbol{x})$ with the logit link function to the predictor

$$\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p, \tag{1}$$

i.e.

$$\text{logit}[\pi(x)] = \ln\left[\frac{\pi(x)}{1-\pi(x)}\right] = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p. \tag{2}$$

Thus

$$\pi(x) = \frac{\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}{1+\exp(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_p x_p)}$$

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

$$\begin{aligned} L(\beta|y) &= \prod_{i=1}^{n}[\pi(x_i)]^{y_i}[1-\pi(x_i)]^{1-y_i} \\ &= \prod_{i=1}^{n}\left[\frac{\exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}{1+\exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}\right]^{y_i} \\ &\times \left[\frac{1}{1+\exp(\beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_p x_{ip})}\right]^{(1-y_i)}, \end{aligned}$$

where $y_i = 0$ or 1. For maximum likelihood estimation, this function can be maximized by taking the natural logrithm of the likelihood function, differentiating with respect to the parameters, equating to zero, solving the equations using the iterative least squares method and obtaining $\hat{\beta_1}, \hat{\beta_2}, \cdots, \hat{\beta_p}$ (McCullagh and Nelder, 1989).

3.1.2 Linear Discriminant Analysis

Although the logistic regression 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 becomes very unstable. Furthermore, for relatively small sample sizes, when the distribution of the features in the model are gaussian distributed, the linear discriminant model (LDA) becomes more stable than the logistic regression 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, $P(\boldsymbol{X}|Y=c)$. LDA assumes that each data class follows or is modeled by a multivariate Gaussian distribution

$$f_c(\boldsymbol{X}) = P(\boldsymbol{X}|Y=c) = \frac{1}{(2\pi)^{d/2}|\Sigma_c|^{1/2}}\exp\bigg(-\frac{1}{2}(\boldsymbol{X}-\boldsymbol{\mu}_c)'\Sigma_c^{-1}(\boldsymbol{X}-\boldsymbol{\mu}_c)\bigg), \tag{3}$$

where d represents the number of features in the model. The covariance matrix Σ_c is the same across all the classes, that is $\Sigma_c = \Sigma$.

As LDA is assumed as a classifier, its use is evidenced by the usage of the class priors estimated from the training data. This is done by finding the prior probabilities, $P(Y = c)$ computed as proportions of data in each class c. The class means μ_c as well as the covariance matrix Σ are estimated by

$$\text{Prior probabilities,}\quad \hat{\pi}_c = \frac{n_c}{n} \tag{4}$$

$$\text{Class means,}\quad \hat{\mu}_c = \frac{1}{n_c}\sum_{i:y_i=c} X_i \tag{5}$$

$$\text{Covariance Matrix,}\ \ \Sigma = \frac{1}{n-C}\sum_{c=1}^{C}\sum_{i:y_i=c}(X_i - \hat{\mu}_k)^2 \tag{6}$$

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

$$q(x) = \arg\max_{c\in R} p_{X|Y=c}(x \mid Y = c)P(Y = c). \tag{7}$$

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

$$d(x) = \begin{cases} 1 & \text{if } P(X|Y=1)P(Y=1) \geq P(X|Y=0)P(Y=0) \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

The general LDA classification function is

$$F(x) = \arg\max_{c} \delta_c(x) \tag{9}$$

where $\delta_c(x)$

$$\delta_c(x) = x'\Sigma^{-1}\mu_c - \frac{1}{2}\mu_c'\Sigma^{-1}\mu_c + \log\pi_c \tag{10}$$

3.1.3 Quadratic Discriminant Analysis

The Linear Discriminant Analysis described in section 3.1.2 classifies/models the binary response with a linear combination of the features. The Quadratic Discriminant Analysis (QDA) is similar to LDA in terms of the derivation of parameters. However, the underlying difference is the 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 cth 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 discrimminant function is

$$\delta_c(x) = -\frac{1}{2}\log|\Sigma_c| - \frac{1}{2}(x-\mu_c)^T\Sigma_c^{-1}(x-\mu_c) + \log\pi_c \tag{11}$$

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.

3.1.4 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 as well. 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 are 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.

3.1.5 Classification Trees

Classification trees are a type of decision tree algorithms. 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 to the region to which it belongs. A classification tree comprises of branches that represent attributes, and leaves representing decisions. In practice, the decision process commences at the trunk and follows the branches until a leaf is reached. For a classification tree algorithm, 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

$$\hat{p}_{sc} = \frac{1}{N_s} \sum_{x_i \in B_s} I(y_i = c). \tag{12}$$

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

$$E = \frac{1}{N_s} \sum_{x_i \in B_s} I(y_i \neq c(s)) = 1 - \hat{p}_{sc(s)}. \tag{13}$$

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

$$\sum_{c \neq c'} \hat{p}_{sc}\hat{p}_{sc'} = \sum_{c=1}^{C} \hat{p}_{sc}(1 - \hat{p}_{sc}). \tag{14}$$

The cross entropy is defined as

$$-\sum_{c=1}^{C} \hat{p}_{sc} \log \hat{p}_{sc}. \tag{15}$$

3.1.6 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, inclusive of 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 are used and tested on multiple models.

The aggregated boostrap or Bagging, is an ensemble method which is an extension of the bootstrap method in maching 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

$$\hat{f}_{\text{bag}}(x) = \frac{1}{G}\sum_{g=1}^{G} \hat{f}^{*g}(x) \tag{16}$$

3.1.7 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 (Hastie et al., 2016) is as follows

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

$$f_0(x) = \arg\min_{\gamma} \sum_{i=1}^{N} L(y_i, \gamma)$$

(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 as pseudo/generalized residuals.

(b) Fit a regression tree to the targets r_{ig} giving terminal regions,

$$R_{jg}, j = 1, 2, \ldots, J_g$$

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

$$\gamma_{jg} = \arg\min_{\gamma} \sum_{x_i \in R_{jg}}^{N} L(y_i, f_{g-1}(x_i) + \gamma)$$

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

    (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},

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

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

3.2 Cross-Validation

After exploring the statistical learning methods and machine learning algorithms presented in section 3.1, it is important that we find an approach to evaluate models. This is where cross-validation comes to play. 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 model's performance throughout the whole dataset. The two cross-validation approaches considered in this project are the validation set approach and the k-fold validation approach.

3.2.1 Validation Set Approach and k-Fold Cross-Validation

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 kickstarter 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 with the validation set approach is its simplicity in terms of implementation and less computational complexity. However, the downside with this method is that it may suffer from issues of high variance. This is as a result of the uncertainty resulting from which observations will end up in either the hold-out 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 project the 5 and 10 folds validation approaches are considered. The merit with this method is its accurate estimation performance. The higher value of k chosen, the less biased model the method results.

3.3 Results

The results of the five machine learning algorithms used to the Kickstarter data and their corresponding test error rates resulting from the cross-validation approaches are tabulated as follows,

Table 3: Evaluation

	Test Error Rate		
Method	VSA	5-Fold CV	10-Fold CV
Logistic Reg	0.06223317	0.05456028	0.05456497
LDA	0.3420085	0.3474692	0.3476556
QDA	0.2825558	0.3007608	0.3009856
Trees	0.0974456	0.1123596	0.1089500
Bagging	0.0075928	0.0055728	0.005343503
Gradient Boosting	0.0250706	0.02555767	0.02517074

Of the 6 methods used to the Kickstarter data, the test error rates obtained across the three cross-validation methods suggest Bagging and Gradient Boosting as being the robust methods in 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 logistic regression model seem 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 $5\% - 6\%$.

4. Discussion and Future Work

This study sought to mainly investigate statistical learning methods and machine learning algorithms that presents 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%) emanated from the United States, the emphasis of the study was placed on these projects. First, a feature engineering was performed in an attempt to target the most relevant variables. After a variable reduction was performed with LASSO, random forests procedure and multicollinearity diagnostics, the variables 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 test error rates were tracked.

The results showed Bagging and Gradient boosting method for classification as having the least test error 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 logistic regression 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 a 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. The Bayesian nonparametric models are more robust and valid as it allows the usage of an infinite number of parameters to capture the information of the distribution underlying the complex data. Moreover, if the interest is the identification of the effect of particular variables considered on the rate of success, then causal inference models rather than curve fitting should be further explored.

REFERENCES

Belleflamme, P., Lambert, T., and Schwienbacher, A. (2014), "Crowdfunding: Tapping The Right Crowd", *Journal of business venturing*, 29(5), 585–609.

Gerber, E. M., and Hui, J. (2014), "Crowdfunding: Motivations and Deterrents for Participation", *ACM Transactions on Computer-Human Interaction*, 20(6), 34–32.

Hastie, T., Tibshirani, R., and Friedman, J. (2016), *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (2nd ed.), Springer.

Hui, J. S., Greenberg, M. D., and Gerber, E. M. (2014, February), Understanding the Role of Community in Crowdfunding Work. In *Proceedings of the 17th ACM conference on Computer Supported Cooperative Work & Social Computing*. DOI: 10.1145/2531602.2531715

Kuppuswamy, V., and Bayus, B. L. (2018), "Crowdfunding Creative Ideas: The Dynamics of Project Backers", *The Economics of Crowdfunding*, D. Cumming, L. Hornuf (eds.), 151–182. Springer.

McCullagh, P., and Nelder, J. A. (1989), *Generalized linear models* (2nd ed.), Chapman and Hall/CRC.

Yu, H., Jiang, S., and Land, K.C. (2015), Multicollinearity in Hierarchical Linear Models. *Social Science Research*, 53 (2015), 118–136.